THE COMPUTATIONAL COMPLEXITY OF CALCULATING PARTITION FUNCTIONS OF OPTIMAL MEDIANS WITH HAMMING DISTANCE

ISTVÁN MIKLÓS[*,†] AND HEATHER SMITH[‡]

Abstract. In this paper, we show that calculating the partition function of optimal medians of binary strings with Hamming distance is #P-complete for several weight functions. The case when the weight function is the factorial function has application in bioinformatics. In that case, the partition function counts the most parsimonious evolutionary scenarios on a star tree under several models in bioinformatics. The results are extended to binary trees and we show that it is also #P-complete to calculate the most parsimonious evolutionary scenarios on an arbitrary binary tree under the substitution model of biological sequences and under the Single Cut-or-Join model for genome rearrangements.

1. The Partition Function

For $n, m \in \mathbb{Z}^+$, fix a multiset of binary strings $B = \{\nu_1, \nu_2, \ldots, \nu_m\}$ where $\nu_i \in \{0,1\}^n$ for each $i \in [m]$. An optimal median μ is a binary string also in $\{0,1\}^n$ which minimizes $\sum_{i=1}^m H(\nu_i, \mu)$ where $H(\nu_i, \mu)$ is the Hamming distance between ν_i and μ. Define $\mathcal{M}(B)$ to be the set of all optimal medians for the multiset B.

Take f to be a non-negative, real-valued function. In this paper, we examine the partition function

$$Z(B, f(x)) = \sum_{\mu \in \mathcal{M}(B)} \prod_{i \in [m]} f(H(\nu_i, \mu)).$$

The case when $f(x) := x!$ has application to phylogenetic trees and genome rearrangement. Under the *Single Cut-or-Join* (Feijão and Meidanis 2011) model for genome rearrangement, genomes are represented as edge labelled directed graphs forming paths and cycles, the direction of the edges along any path and cycle might vary. Such a graph can be encoded in binary strings, where each possible pair of edge endpoints (adjacency) is represented with one bit. The bit is 1 if the adjacency is presented in the genome and 0 otherwise. Note that although any genome can be represented with such a binary string, not all binary strings represent a genome since two adjacencies might be in conflict if they share a common edge endpoint, and thus, their bits cannot be both 1. A mutation is a bit flip: a flip from 0 to 1 represents a join, a flip from 1 to 0 represents a cut. Any flip from 1 to 0 is possible, however, flipping 0 to 1 is possible only if it does not cause a conflict. Still, it can be proved that if two binary strings μ and ν represent two genomes G_1 and G_2 under the Single Cut-or-Join model, the fewest number of mutations to transform G_1 into G_2 is $H(\mu, \nu)$ (Feijão and Meidanis 2011). A *scenario* is an ordering of the mutations necessary such that each intermediate string obtained from performing the cuts and joins one at a time represents a valid genome. An upper bound on the number of scenarios is $H(\mu, \nu)!$. This upper bound is precisely the number of scenarios if there is no conflict in the presented adjacencies in genomes G_1 and G_2; that is, there is no constraint that some of the adjacencies first must be cut before some other adjacencies are created with joins.

Next fix a multiset, B, of m binary strings from $\{0,1\}^n$. If these strings label the leaves of a star tree $K_{1,m}$, the center of the star, or common ancestor, should be labeled with a median from $\mathcal{M}(B)$ which minimizes the number of mutations required, summed over all edges of the star tree. A *most parsimonious scenario* for $K_{1,m}$ with B labeling the leaves consists of a median μ from $\mathcal{M}(B)$ and a scenario transforming μ to ν_i for each $i \in [m]$ to label the edges of the star tree. The partition function $Z(B, x!)$ counts the number of most parsimonious scenarios if there is no conflict in the presented adjacencies. In this paper, we show that counting the most parsimonious scenarios is computationally hard already for these special cases.

[*] MTA Rényi Institute, Reáltanoda u. 13-15, 1053 Budapest, Hungary
[†] MTA SZTAKI, Lágymányosi u. 11, 1111 Budapest, Hungary
[‡] School of Mathematics, Georgia Institute of Mathematics, Atlanta, GA 30332, USA
Email: miklos.istvan@renyi.mta.hu (István Miklós) heather.smith@math.gatech.edu (Heather Smith).

Other bioinformatics models also use strings and changes of characters in them, for example when the strings represent biological sequences (DNA, RNA or protein) and changes of characters represent substitutions. We will refer to these models as *substitution models* of biological sequences (Felsenstein 2003). The negative results represented in this paper also holds for these models, too.

In Section 2, we establish some basics about computational complexity classes. Sections 3 and 4 are devoted to computing the value of $Z(B, x!)$. In Section 5, we explore the possibility of stochastic approximations for $Z(B, x!)$. Then we turn our attention to the more general $Z(B, f(x))$ in Section 6. When $\log f(x)$ is strictly concave up or strictly concave down, we obtain some further computational complexity results under mild restrictions. Section 7 is devoted to stochastic approximations for $Z(B, f(x))$ when $\log f(x)$ is strictly concave down. In Sections 8 and 9, we extend our exploration of $Z(B, x!)$ from star trees to binary trees and define a similar partition function.

2. Computational complexity

While P and NP are complexity classes for decision problems, the following classes are for counting problems. The classes #P, #P-hard, and #P-complete were first defined by Valiant (1979). The definition for #P that we give here, while not the original, is an equivalent definition.

Definition 2.1 (Welsh 1993). *The class* #P *contains those functions* $f : \Sigma^* \to \mathbb{N}$*, for some alphabet* Σ*, such that both of the following hold:*

- *There is a polynomial* p*, a relation* R*, and a polynomial time algorithm which, for each input* $w \in \Sigma^*$ *and each* $y \in \Sigma^*$ *with* $|y| \leq p(|w|)$*, determines if* $R(w, y)$*.*
- *For any input* w*,* $f(w) = |\{y : |y| \leq p(|w|) \text{ and } R(w, y)\}|$*.*

Definition 2.2 (Valiant 1979). *A counting problem is in* #P-hard *if there is a polynomial time reduction to it from every problem in #P. A counting problem is in* #P-complete *if it is in #P and is in #P-hard.*

Next we give a few known computational complexity results. To state these result, we establish some terminology.

Conjunctive normal form (CNF) is a standard format in which to express Boolean formulas. A *3CNF* is a Boolean formula Γ which is the conjunction of clauses and each clause is the disjunction of 3 literals. A 3CNF, Γ, with n variables $\{v_1, v_2, \ldots, v_n\}$ and k clauses takes the form $\Gamma = c_1 \wedge c_2 \wedge \ldots \wedge c_k$ where each c_i is a clause which is the disjunction of three literals and the literals are from $\{v_i\}_{i=1}^n \cup \{\overline{v_i}\}_{i=1}^n$. Because Γ was said to have n variables, we may assume that, for each $i \in [n]$, v_i or $\overline{v_i}$ appears in some clause of Γ. Each v_i is a *positive literal* while each $\overline{v_i}$ is a *negative literal*. The negative literal $\overline{v_i}$ is the negation of v_i. We identify $\overline{\overline{v_i}}$ with the literal v_i and we refer to $\{v_i\}_{i=1}^m$ as the *variables* of Γ and always assume that the set of variables has an ordering.

A *truth assignment* for Γ is a function $f : \{v_i\}_{i=1}^n \to \{T, F\}^n$ which assigns a value of true or false to each variable. If a truth assignment makes Γ true, we say it satisfies Γ. Otherwise, a truth assignment does not satisfy Γ in which case there is at least one clause which is not satisfied.

Definition 2.3 (3SAT). *Given an arbitrary* Γ *in 3CNF with* n *variables and* k *clauses, decide if there is a truth assignment for* Γ *which satisfies* Γ*.*

Definition 2.4 (#3SAT). *Given an arbitrary Boolean formula* Γ *in 3CNF with* n *variables and* k *clauses, count the number of truth assignments which satisfy* Γ*.*

Theorem 2.5 (Cook 1971). *3SAT* $\in$ *NP-complete.*

Theorem 2.6 (Valiant 1979). *#3SAT* $\in$ *#P-complete.*

Define *D3CNF* to be the subset of 3CNF containing only those $\Gamma = \bigwedge_{i \in [k]} c_i$ such that for each $i \in [k]$,

- c_i contains three distinct literals, and
- c_i does not contain both v_j and $\overline{v_j}$ for any $j \in [n]$.

This defines the following two problems.

Definition 2.7 (D3SAT). *For an arbitrary* Γ *in D3CNF with* n *variables and* k *clauses, decide if there is a truth assignment which satisfies* Γ*.*

Definition 2.8 (#D3SAT)**.** *For an arbitrary Γ in D3CNF with n variables and k clauses, count the number of truth assignments which satisfy Γ.*

The following two results are proven through reductions from #3SAT and 3SAT.

Lemma 2.9. *#D3SAT $\in$ #P-complete.*

Proof. This is a reduction from #3SAT. Let Γ be a 3CNF with n variables and k clauses, $n \geq 3$. Let $v_\alpha, v_\beta, v_\gamma$ be literals in Γ with $\alpha \neq \beta \neq \gamma \neq \alpha$. Observe that each of the following pairs have the same satisfying truth assignments.

$$(v_\alpha \vee v_\beta \vee v_\beta) \text{ and } (v_\alpha \vee v_\beta \vee v_\gamma) \wedge (v_\alpha \vee v_\beta \vee \overline{v_\gamma}).$$
$$(v_\alpha \vee v_\alpha \vee v_\alpha) \text{ and } (v_\alpha \vee v_\beta \vee v_\gamma) \wedge (v_\alpha \vee \overline{v_\beta} \vee v_\gamma) \wedge (v_\alpha \vee v_\beta \vee \overline{v_\gamma}) \wedge (v_\alpha \vee \overline{v_\beta} \vee \overline{v_\gamma}).$$

Further, a clause of the form $(v_\alpha \vee \overline{v_\alpha} \vee v_\beta)$ is alway true, so it can be removed.

Making these replacements in Γ will result in a D3CNF Γ' with n' variables ($n' \leq n$) and at most $4k$ clauses. Because some clauses like $(v_\alpha \vee \overline{v_\alpha} \vee v_\beta)$ are in Γ but not in Γ', it is possible that $n' < n$.

Given a satisfying truth assignment for Γ', we may extend it to a satisfying truth assignment for Γ in $2^{n-n'}$ ways. This is because the variables in Γ which are not in Γ' do not affect the ability of a truth assignment to satisfy Γ. On the other hand, each satisfying truth assignment for Γ, restricted to the variables of Γ', will be a satisfying truth assignment for Γ'. □

Lemma 2.10. *D3SAT $\in$ NP-complete.*

Proof. As described in the last proof, for any 3CNF Γ, there is a corresponding D3CNF Γ' which is computable in polynomial time such that Γ' has at least one satisfying truth assignment exactly when Γ has at least one satisfying truth assignment. □

Next, we return our attention to the partition functions which were introduced in Section 1. The complexity results in this paper address subquestions and analogues of the following problems.

Definition 2.11 (#SPS)**.** *Given a tree T and a labeling φ of the leaves of T with binary strings, #SPS (*Small Parsimony Substitution*) asks for the exact number of most parsimonious scenarios where the scenario on an edge is an ordering of the substitutions that must take place to transform the median sequence into the sequence at the leaf.*

Definition 2.12 (#SPSCJ)**.** *Given a tree T and a labeling φ of the leaves of T with binary strings representing genomes under the SCJ model, #SPSCJ (*Small Parsimony Single Cut-or-Join*) asks for the exact number of most parsimonious scenarios where the scenario on an edge is an ordering of the cuts and joins that must take place to transform the median sequence into the sequence at the leaf such that the sequence produced after each cut or join represents a valid genome.*

Clearly, #SPS is a special case of #SPSCJ, the case when there is no conflict in the adjacencies present in the genomes assigned to the leaves of the evolutionary tree. While $Z(B, x!)$ is only an upper bound for an instance of #SPSCJ, it is the exact answer to each instance of #SPS.

Lemma 2.13. *The problem of calculating $Z(B, x!)$ is in #P.*

Proof. The input includes a multiset $B = \{\nu_i\}_{i=1}^m$ where each ν_i is from $\{0,1\}^n$. Viewing this in the sense of phylogenetic trees, a witness consists of a median μ to label the center of the star and a scenario to label each edge of the tree. The size of the input is $O(mn)$.

Recall that a median μ' minimizes the quantity $\sum_{i=1}^m H(\nu_i, \mu')$. We can find a single median μ' in $O(mn)$ time by examining the k^{th} coordinate of each string in B and making the k^{th} coordinate of μ' the value that appears in a majority of the strings in B, breaking ties arbitrarily. To verify that the given binary string μ is indeed a median, we need only compare $\sum_{i=1}^m H(\nu_i, \mu)$ and $\sum_{i=1}^m H(\nu_i, \mu')$. If they are the same, then μ is a median. For each edge, we can verify that the given permutation is a scenario for that edge in $O(m)$ time by comparing the bits of μ and ν_i. By Definition 2.1, #SPS is in #P. □

Most of the complexity results are reductions from #D3SAT. In other words, given a D3CNF Γ with n variables and k clauses, we create a multiset of m binary strings of length $2n + t$ (where t and m are polynomials of n and k) to label the leaves of the tree. These strings will be chosen so that the number of most parsimonious substitution scenarios is related to the number of satisfying truth assignments for Γ.

3. Set-up for results on star trees

The discussion in this section and the next is specific to $Z(B, x!)$. The first section details some tools and constructions that will be needed for the proof of our main result in Section 4. This main result, Theorem 4.4, states that computing $Z(B, x!)$ is #P-complete.

The proof of Theorem 4.4 will define a polynomial reduction from #D3SAT (Definition 2.8) to compute $Z(B, x!)$. Fix an arbitrary D3CNF, Γ, with n variables and k clauses. Fix a prime $p \leq 5\max\{300, n+5\}$ which will be utilized later.

Our task is to define a multiset of binary strings $\mathcal{D}(p)$ to encode Γ. The multiset $\mathcal{D}(p)$ will be chosen so that the set of medians $\mathcal{M}(\mathcal{D}(p))$ will have a list of desired characteristics. First, each string in $\mathcal{D}(p)$ and each median in $\mathcal{M}(\mathcal{D}(p))$ will have length $2n + t$ with coordinates

$$(x_1, y_1, x_2, y_2, \ldots, x_n, y_n, e_1, e_2, \ldots, e_t)$$

where n is the number of variables in Γ and the t is a polynomial of n and k which will be defined later. Second, $\mathcal{M}(\mathcal{D}(p))$ will be the set of all binary strings μ of length $2n+t$ that have $\mu[e_i] = 0$ for each $i \in [t]$. In other words, $\mathcal{D}(p)$ will be defined so that $\mathcal{M}(\mathcal{D}(p))$ equals $\{0,1\}^{2n} \times \{0\}^t$. Let $\mathcal{M}'(\mathcal{D}(p))$ denote the subset of $\mathcal{M}(\mathcal{D}(p))$ with the additional property that $\mu[x_i] \neq \mu[y_i]$ for all $i \in [n]$. Once we have established that $\mathcal{M}(\mathcal{D}(p)) = \{0,1\}^{2n} \times \{0\}^t$, we can conclude $\mathcal{M}'(\mathcal{D}(p)) = \{01, 10\}^n \times \{0\}^t$. This allows for a connection with truth assignments for Γ.

Definition 3.1. *Let $n \in \mathbb{Z}^+$. For arbitrary Γ in D3CNF with n variables, let S be a multiset of binary strings on the coordinates $(x_1, y_1, \ldots, x_n, y_n, e_1, \ldots, e_t)$. There is an injective function f which assigns to each median $\mu \in \mathcal{M}'(S)$ a truth assignment for Γ. In particular, $f(\mu)$ will assign a value of true to the i^{th} variable of Γ if $\mu[x_i] = 1$ and false if $\mu[x_i] = 0$.*

Remark 3.2. *If multiset S is chosen so that $\mathcal{M}'(S) = \{01, 10\}^n \times \{0\}^t$, then Definition 3.1 provides a bijection between $\mathcal{M}'(S)$ and the truth assignments for Γ.*

Definition 3.3. *Let $n \in \mathbb{Z}^+$. Given an arbitrary D3CNF, Γ, with n variables, let S be an arbitrary multiset of binary strings on the coordinates $(x_1, y_1, \ldots, x_n, y_n, e_1, \ldots, e_t)$ for some $t \in \mathbb{Z}^+$. Define $\mathcal{M}'_\Gamma(S)$ to be a subset of $\mathcal{M}'(S)$, containing only those medians which, through the bijection in Definition 3.1, correspond to a satisfying truth assignment for Γ. Since a single clause c in Γ is also a D3CNF, this defines $\mathcal{M}'_c(S)$ as well.*

To calculate

$$Z(\mathcal{D}(p), x!) = \sum_{\mu \in \mathcal{M}(\mathcal{D}(p))} \prod_{i \in [m]} H(\mu, \nu_i)!,$$

we first calculate $\prod_{i \in [m]} H(\mu, \nu_i)!$ for each median $\mu \in \mathcal{M}(\mathcal{D}(p))$. The multiset $\mathcal{D}(p)$ will be constructed so that there is a constant $K(p)$ (specified in Claim 4.8) which is not a multiple of p and such that for any $\mu \in \mathcal{M}'_\Gamma(\mathcal{D}(p))$ $\prod_{i \in [m]} H(\mu, \nu_i)! = K(p)$. Each median $\mu \in \mathcal{M}(\mathcal{D}(p)) \setminus \mathcal{M}'_\Gamma(\mathcal{D}(p))$ will have $\prod_{i \in [m]} H(\mu, \nu_i)! \equiv 0 \mod p$. As a result

$$\sum_{\mu \in \mathcal{M}(\mathcal{D}(p))} \prod_{i \in [m]} H(\mu, \nu_i)! \equiv |\mathcal{M}'_\Gamma(\mathcal{D}(p))| K(p) \mod p.$$

Repeating this construction for sufficiently many primes $p \leq 5\max\{300, n+5\}$, we obtain enough congruences, which together with the knowledge that there are at most 2^n satisfying truth assignment for Γ, uniquely determine the size of $\mathcal{M}'_\Gamma(\mathcal{D}(p))$ which is equal to the number of satisfying truth assignments for Γ.

Later we will see that the main work goes into developing a multiset $\mathcal{D}(p)$ with the property that for any $\mu \in \mathcal{M}'_\Gamma(\mathcal{D}(p))$ and any $\mu' \in \mathcal{M}'(\mathcal{D}(p)) \setminus \mathcal{M}'_\Gamma(\mathcal{D}(p))$ have

$$\prod_{i \in [m]} H(\mu, \nu_i)! \neq \prod_{i \in [m]} H(\mu', \nu_i)!.$$

In Section 3.1, we define the strings $\mathcal{D}(p)$ which are used in the proofs to distinguish medians in $\mathcal{M}'_\Gamma(\mathcal{D}(p))$ from medians in $\mathcal{M}'(\mathcal{D}(p)) \setminus \mathcal{M}'_\Gamma(\mathcal{D}(p))$.

3.1. **Encoding Boolean clauses in binary strings.**

A truth assignment satisfies Γ if and only if it satisfies every clause in Γ. Hence, we will encode each clause c_i of Γ in a set of 50 strings

$$\mathcal{C}_i := \{\nu_1^i, \nu_2^i, \dots, \nu_{50}^i\}$$

which will be defined through Table 1. These 50 strings are designed to distinguish those medians in $\mathcal{M}'_{c_i}(\mathcal{C}_i)$ from those in $\mathcal{M}'(\mathcal{C}_i)\backslash\mathcal{M}'_{c_i}(\mathcal{C}_i)$. Confirmation of this will come in Section 3.3. Because every truth assignment that does not satisfy Γ has at least one clause in Γ that is does not satisfy, we will see that the disjoint union $\biguplus_{i\in[k]}\mathcal{C}_i$ distinguishes between $\mathcal{M}'_\Gamma\left(\biguplus_{i\in[k]}\mathcal{C}_i\right)$ and $\mathcal{M}'\left(\biguplus_{i\in[k]}\mathcal{C}_i\right)\setminus\mathcal{M}'_\Gamma\left(\biguplus_{i\in[k]}\mathcal{C}_i\right)$.

The following definition gives a guide for defining a multiset of binary strings.

Definition 3.4 (Defining strings). *For arbitrary $m, n \in \mathbb{Z}^+$ and $t \in \mathbb{Z}^+ \cup \{0\}$, to define a multiset of binary strings $\{\eta_1, \eta_2, \dots, \eta_m\}$ on coordinates $(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t)$, it suffices to*

- *define $\eta_j[x_\ell]$ and $\eta_j[y_\ell]$ for each $j \in [m]$ and $\ell \in [n]$, and*
- *define a function $e : [m] \to \mathbb{Z}^+ \cup \{0\}$.*

We say η_j has $e(j)$ additional ones. In order to infer the values $\eta_j[e_\ell]$ for each $j \in [m]$ and $\ell \in [t]$, follow this procedure:

> *Partition $[t]$ into subsets $E, E_1, E_2, \dots, E_m$ so that the size of E_j is precisely $e(j)$, and $E = [t] \setminus \bigcup_{j\in[m]} E_j$. For each $j \in [m]$ and each $\ell \in [t]$, set $\eta_j[e_\ell] = 1$ if and only if $\ell \in E_j$ and $\eta_{j'}[e^\ell] = 0$ otherwise.*

Remark 3.5. *Let $m \in \mathbb{Z}^+$. For an arbitrary multiset $\{\eta_j\}_{j=1}^m$ of binary strings built using Definition 3.4, for each $\ell \in [t]$, there is a unique $j \in [m]$ such that $\eta_j[e_\ell] = 1$. Consequently, each $\mu \in \mathcal{M}(\{\eta_j\}_{j=1}^m)$ will have $\mu[e_\ell] = 0$ for all $\ell \in [t]$ because μ must minimize $\sum_{j\in[m]} H(\eta_j, \mu)$.*

Definition 3.6. *Let $n \in \mathbb{Z}^+$ and $t \in \mathbb{Z}^+ \cup \{0\}$ be arbitrary. Two binary strings η and $\overline{\eta}$ with coordinates $(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t)$, are said to be* complementary on the first $2n$ coordinates *if $\eta[x_i] = 1 - \overline{\eta}[x_i]$ and $\eta[y_i] = 1 - \overline{\eta}[y_i]$ for each $i \in [n]$.*

The following fact will be useful.

Fact 3.7. *Let η and $\overline{\eta}$ be binary strings on coordinates*

$$(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t).$$

Set $e(\eta) := \sum_{i\in[t]} \eta[e_i]$, the number of additional ones in η. Define $e(\overline{\eta})$ similarly. If η and $\overline{\eta}$ are complementary on the first $2n$ coordinates, then for any $\mu \in \{0,1\}^{2n} \times \{0\}^t$,

$$H(\mu, \eta) + H(\mu, \overline{\eta}) = 2n + e(\eta) + e(\overline{\eta}).$$

Proof. For each $i \in [n]$, either $\mu[x_i] = \eta[x_i]$ or $\mu[x_i] = \overline{\eta}[x_i]$, but not both. This is also true for each y_i. This accounts for the $2n$ in the sum. Because $\mu[e_i] = 0$ for all $i \in [t]$, each $i \in [t]$ with $\eta[e_i] = 1$ will contribute one to the sum. Also each $i \in [t]$ with $\overline{\eta}[e_i]$ will contribute one to the sum. This completes the proof. □

Definition 3.8. *Given an arbitrary multiset of binary strings S, we say that a coordinate s is* ambiguous *if there are exactly $\frac{1}{2}|S|$ binary strings $\eta \in S$, counted with multiplicity, such that $\eta[s] = 0$. Consequently, if you change the value of a median at an ambiguous coordinate, you obtain another median. Note that if $|S|$ is odd, then there are no ambiguous coordinates and there is exactly one median.*

Fact 3.9. *Let S be a multiset of binary strings which are defined on the coordinates*

$$(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t).$$

If S can be partitioned into pairs of strings where the two strings in a pair are complementary on the first $2n$ coordinates, then each x_i and each y_i is an ambiguous coordinate.

Fix an arbitrary D3CNF, Γ, with n variables and k clauses. Fix a clause c_i in Γ. For this clause, we are now ready to define a set of 50 strings

$$\mathcal{C}_i = \{\nu_1^i, \nu_2^i, \dots, \nu_{50}^i\}.$$

First assume that $c_i = v_\alpha \vee v_\beta \vee v_\gamma$, a disjunction of three positive literals. Because Γ is a D3CNF, we may assume $\alpha < \beta < \gamma$.

For each $j \in [50]$, we will supply the following three pieces of information for ν_j^i:

(a) The values for $\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma]$ will be explicitly defined.
(b) A constant $\kappa_{ij} \in \{0,1\}$ will be given so that $\nu^i_j[x_\ell] = \nu^i_j[y_{\ell'}] = \kappa_{ij}$ for all $\ell, \ell' \in [n] \setminus \{\alpha, \beta, \gamma\}$.
(c) The string will be assigned some number of additional ones.

By Definition 3.4, this is sufficient to explicitly define ν^i_j.

In Table 1, there is a row for each string in $\mathcal{C}_i$. The three defining pieces of information are found in Columns A, B, and C respectively. The remainder of the table will be explained in Subsection 3.2.

For each $j \in [50]$, row j of Table 1 supplies the three ingredients needed to define ν^i_j. By matching the 6-bit string in Column A of row j with

$$(\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma])$$

we obtain the 6 values for (a). The constant κ_{ij} for (b) is found in Column B of row j. For (c), the number of additional ones in ν^i_j is found in Column C of row j.

With a slight modification in the reading of Column A, the 50 rows of Table 1 will also supply the 50 strings for a clause which contains negative literals. Fix an arbitrary clause c_i in Γ which now may have negative literals. For each $j \in [50]$, the definition of string ν^i_j will again be based on Columns A, B, C of row j in Table 1 where the same information will be gleaned from Columns B and C. The only difference is with Column A which will be explained next.

If c_i contains the variables $v_\alpha, v_\beta, v_\gamma$ where some of these may be present as negative literals, set $S_i := \{x_\alpha, y_\alpha, x_\beta, y_\beta, x_\gamma, y_\gamma\}$. We call S_i the *support set* of c_i. Clause c_i must be one of the 8 clauses listed in Column A of Table 2. For $j \in [50]$, ν^i_j is defined on the coordinates S_i by matching the entry in the right column of the c_i row of Table 2 with the 6-bit string in Column A of the j^{th} row of Table 1.

Example 3.10. *For an example, when $c_i = v_\alpha \vee \overline{v_\beta} \vee \overline{v_\gamma}$, the last row of Table 1 says that the string ν^i_{50} must have*

$$(\nu^i_{50}[x_\alpha], \nu^i_{50}[y_\alpha], \nu^i_{50}[y_\beta], \nu^i_{50}[x_\beta], \nu^i_{50}[y_\gamma], \nu^i_{50}[x_\gamma]) = (101010).$$

Therefore, $\nu^i_{50}[x_\alpha] = 1$, $\nu^i_{50}[y_\alpha] = 0$, $\nu^i_{50}[x_\beta] = 0$, $\nu^i_{50}[y_\beta] = 1$, $\nu^i_{50}[x_\gamma] = 0$, and $\nu^i_{50}[y_\gamma] = 1$. Further, Column B implies $\nu^i_{50}(x_\ell) = \nu^i_{50}(y_\ell) = 1$ for all $\ell \in [n] \setminus \{\alpha, \beta, \gamma\}$ and, from Column C, ν^i_{50} will have 2 additional ones.

Now that we have defined $\mathcal{C}_i$ for any clause c_i, let us analyze $\mathcal{M}(\mathcal{C}_i)$. By Fact 3.5, for every $\mu \in \mathcal{M}(\mathcal{C}_i)$ and $\ell \in [t]$, $\mu[e_\ell] = 0$.

In Column B of Table 1, it is evident that for any $\ell \in [n] \setminus \{\alpha, \beta, \gamma\}$, the number of strings ν^i_j with $\nu^i_j[x_\ell] = 0$ is $25 = \frac{1}{2}|\mathcal{C}_i|$. Therefore, by Definition 3.8, the coordinates x_ℓ and y_ℓ are ambiguous. Through careful inspection of the strings in Column A of Table 1, we see that coordinates $x_{\ell'}$ and $y_{\ell'}$ are also ambiguous for each $\ell' \in \{\alpha, \beta, \gamma\}$. Therefore we have proven the following fact, which was one of our goals:

Fact 3.11. *For an arbitrary clause c_i with three distinct variables,*

$$\mathcal{M}(\mathcal{C}_i) = \{0,1\}^{2n} \times \{0\}^t.$$

Remark 3.12. *By visual inspection of Table 1, the binary strings $\mathcal{C}_i$ can be partitioned into pairs where the two strings in a pair are complementary on the first $2n$ coordinates.*

3.2. **Hamming distances between C_i and possible medians.**

Here we explain the remainder of Table 1. Fix a clause c_i in Γ which will be used throughout this subsection. Suppose c_i has variables v_α, v_β, and v_γ. By Fact 3.11, $\mathcal{M}(\mathcal{C}_i) = \{0,1\}^{2n} \times \{0\}^t$. Therefore, $\mathcal{M}'(\mathcal{C}_i)$ must be equal to $\{01, 10\}^n \times \{0\}^t$. For this subsection, define

$$\mathcal{M} := \mathcal{M}(\mathcal{C}_i), \qquad \mathcal{M}' := \mathcal{M}'(\mathcal{C}_i).$$

Define an equivalence relation $\sim_i$ on $\mathcal{M}'$ such that two medians are equivalent if they agree on the coordinates in the support set S_i of c_i. The result will be 8 equivalence classes because $\mu[x_\ell] \neq \mu[y_\ell]$ for each $\ell \in \{\alpha, \beta, \gamma\}$ for each $\mu \in \mathcal{M}'$.

Here we define a one-to-one correspondence between the equivalence classes of $\mathcal{M}'$ under $\sim_i$ and the 6-bit strings heading Columns M1 through M8 in Table 1.

Table 1. The 50 strings in $\mathcal{C}_i$ for a single clause c_i along with their Hamming distance from medians in $\mathcal{M}'$.

		A	**B**	**C**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
	Row #	Values of ν_j^i on its support set	$\nu_j^i[x_\ell]$, $\nu_j^i[y_\ell]$ $(v_\ell \notin c_i)$	Add'l Ones	10 10 10	10 10 01	10 01 10	01 10 10	10 01 01	01 10 01	01 01 10	01 01 01
$\mathcal{N}_1^{(+3)}$	1	01 00 00	0	+3	$n+4$	$n+4$	$n+4$	$n+2$	$n+4$	$n+2$	$n+2$	$n+2$
	2	00 01 00	0	+3	$n+4$	$n+4$	$n+2$	$n+4$	$n+2$	$n+4$	$n+2$	$n+2$
	3	00 00 01	0	+3	$n+4$	$n+2$	$n+4$	$n+4$	$n+2$	$n+2$	$n+4$	$n+2$
$\overline{\mathcal{N}_1}^{(+0)}$	4	10 11 11	1	+0	$n-1$	$n-1$	$n-1$	$n+1$	$n-1$	$n+1$	$n+1$	$n+1$
	5	11 10 11	1	+0	$n-1$	$n-1$	$n+1$	$n-1$	$n+1$	$n-1$	$n+1$	$n+1$
	6	11 11 10	1	+0	$n-1$	$n+1$	$n-1$	$n-1$	$n+1$	$n+1$	$n-1$	$n+1$
$\mathcal{I}_2^{(+2)} \setminus \mathcal{N}_2^{(+2)}$	7	10 10 00	0	+2	n	n	$n+2$	$n+2$	$n+2$	$n+2$	$n+4$	$n+4$
	8	10 00 10	0	+2	n	$n+2$	n	$n+2$	$n+2$	$n+4$	$n+2$	$n+4$
	9	00 10 10	0	+2	n	$n+2$	$n+2$	n	$n+4$	$n+2$	$n+2$	$n+4$
	10	10 10 00	0	+2	n	n	$n+2$	$n+2$	$n+2$	$n+2$	$n+4$	$n+4$
	11	10 00 01	0	+2	$n+2$	n	$n+2$	$n+4$	n	$n+2$	$n+4$	$n+2$
	12	00 10 01	0	+2	$n+2$	n	$n+4$	$n+2$	$n+2$	n	$n+4$	$n+2$
	13	10 01 00	0	+2	$n+2$	$n+2$	n	$n+4$	n	$n+4$	$n+2$	$n+2$
	14	10 00 10	0	+2	n	$n+2$	n	$n+2$	$n+2$	$n+4$	$n+2$	$n+4$
	15	00 01 10	0	+2	5	$n+4$	n	$n+2$	$n+2$	$n+4$	n	$n+2$
	16	01 10 00	0	+2	$n+2$	$n+2$	$n+4$	n	$n+4$	n	$n+2$	$n+2$
	17	01 00 10	0	+2	$n+2$	$n+4$	$n+2$	n	$n+4$	$n+2$	n	$n+2$
	18	00 10 10	0	+2	n	$n+2$	$n+2$	n	$n+4$	$n+2$	$n+2$	$n+4$
	19	10 01 00	0	+2	$n+2$	$n+2$	n	$n+4$	n	$n+4$	$n+2$	$n+2$
	20	10 00 01	0	+2	$n+2$	n	$n+2$	$n+4$	n	$n+2$	$n+4$	$n+2$
	21	00 01 01	0	+2	$n+4$	$n+4$	$n+2$	$n+4$	n	$n+2$	$n+2$	n
	22	01 10 00	0	+2	$n+2$	$n+2$	$n+4$	n	$n+4$	n	$n+2$	$n+2$
	23	01 00 01	0	+2	$n+4$	$n+2$	$n+4$	$n+2$	$n+2$	n	$n+2$	n
	24	00 10 01	0	+2	$n+2$	n	$n+4$	$n+2$	$n+2$	n	$n+4$	$n+2$
	25	01 01 00	0	+2	$n+4$	$n+4$	$n+2$	$n+2$	$n+2$	$n+2$	n	n
	26	01 00 10	0	+2	$n+2$	$n+4$	$n+2$	n	$n+4$	$n+2$	n	$n+2$
	27	00 01 10	0	+2	$n+2$	$n+4$	n	$n+2$	$n+2$	$n+4$	n	$n+2$
$\overline{\mathcal{I}_2}^{(+1)} \setminus \overline{\mathcal{N}_2}^{(+1)}$	28	10 10 11	1	+1	$n-1$	$n-1$	$n+1$	$n+1$	$n+1$	$n+1$	$n+3$	$n+3$
	29	10 11 01	1	+1	$n+1$	$n-1$	$n+1$	$n+3$	$n-1$	$n+1$	$n+3$	$n+1$
	30	11 10 01	1	+1	$n+1$	$n-1$	$n+3$	$n+1$	$n+1$	$n-1$	$n+3$	$n+1$
	31	10 01 11	1	+1	$n+1$	$n+1$	$n-1$	$n+3$	$n-1$	$n+3$	$n+1$	$n+1$
	32	10 11 10	1	+1	$n-1$	$n+1$	$n-1$	$n+1$	$n+1$	$n+3$	$n+1$	$n+3$
	33	11 01 10	1	+1	$n+1$	$n+3$	$n-1$	$n+1$	$n+1$	$n+3$	$n-1$	$n+1$
	34	01 10 11	1	+1	$n+1$	$n+1$	$n+3$	$n-1$	$n+3$	$n-1$	$n+1$	$n+1$
	35	01 11 10	1	+1	$n+1$	$n+3$	$n+1$	$n-1$	$n+3$	$n+1$	$n-1$	$n+1$
	36	11 10 10	1	+1	$n-1$	$n+1$	$n+1$	$n-1$	$n+3$	$n+1$	$n+1$	$n+3$
	37	10 01 11	1	+1	$n+1$	$n+1$	$n-1$	$n+3$	$n-1$	$n+3$	$n+1$	$n+1$
	38	10 11 01	1	+1	$n+1$	$n-1$	$n+1$	$n+3$	$n-1$	$n+1$	$n+3$	$n+1$
	39	11 01 01	1	+1	$n+3$	$n+1$	$n+1$	$n+3$	$n-1$	$n+1$	$n+1$	$n-1$
	40	01 10 11	1	+1	$n+1$	$n+1$	$n+3$	$n-1$	$n+3$	$n-1$	$n+1$	$n+1$
	41	01 11 01	1	+1	$n+3$	$n+1$	$n+3$	$n+1$	$n+1$	$n-1$	$n+1$	$n-1$
	42	11 10 01	1	+1	$n+1$	$n-1$	$n+3$	$n+1$	$n+1$	$n-1$	$n+3$	$n+1$
	43	01 01 11	1	+1	$n+3$	$n+3$	$n+1$	$n+1$	$n+1$	$n+1$	$n-1$	$n-1$
	44	01 11 10	1	+1	$n+1$	$n+3$	$n+1$	$n-1$	$n+3$	$n+1$	$n-1$	$n+1$
	45	11 01 10	1	+1	$n+1$	$n+3$	$n-1$	$n+1$	$n+1$	$n+3$	$n-1$	$n+1$
	46	01 01 11	1	+1	$n+3$	$n+3$	$n+1$	$n+1$	$n+1$	$n+1$	$n-1$	$n-1$
	47	01 11 01	1	+1	$n+3$	$n+1$	$n+3$	$n+1$	$n+1$	$n-1$	$n+1$	$n-1$
	48	11 01 01	1	+1	$n+3$	$n+1$	$n+1$	$n+3$	$n-1$	$n+1$	$n+1$	$n-1$
$\mathcal{N}_3^{(+1)}$	49	01 01 01	0	+1	$n+3$	$n+2$	$n+2$	$n+4$	n	n	n	$n-2$
$\overline{\mathcal{N}_3}^{(+2)}$	50	10 10 10	1	+2	$n-1$	$n+1$	$n+1$	$n+1$	$n+3$	$n+3$	$n+3$	$n-3$

For a clause c_i, the left three columns define the 50 strings in $\mathcal{C}_i$. In row j, the 6-bit string gives the values of ν_j^i on the support set S_i as described by Table 2. The second column gives the constant value to be assigned to all x_ℓ and y_ℓ which are not in S_i. The third column specifies the number of extra ones in ν_j^i. The collection $\{01, 10\}^3$ is listed along the top row. The entry in row j and column ℓ is the number of additional ones in ν_j^i added to the Hamming distance between the 6-bit string in row j and the 6-bit string at the top of column ℓ.

Definition 3.13. *Fix a clause c_i and an integer $\ell \in [8]$. Consider the 6-bit string δ which heads column Mℓ. In Table 2, locate the tuple in the right column corresponding to our fixed clause c_i. After replacing each ν_j^i with μ in the tuple, match this tuple with δ. This gives six values that a median $\mu \in \mathcal{M}'$ must have if it is in the equivalence class represented by the column heading δ.*

Table 2. A key for interpreting Column A of Table 1.

Clause	Key to interpret Column A of Table 1
$v_\alpha \vee v_\beta \vee v_\gamma$	$(\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma])$
$\overline{v_\alpha} \vee v_\beta \vee v_\gamma$	$(\nu^i_j[y_\alpha], \nu^i_j[x_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma])$
$v_\alpha \vee \overline{v_\beta} \vee v_\gamma$	$(\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[y_\beta], \nu^i_j[x_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma])$
$v_\alpha \vee v_\beta \vee \overline{v_\gamma}$	$(\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[y_\gamma], \nu^i_j[x_\gamma])$
$\overline{v_\alpha} \vee \overline{v_\beta} \vee v_\gamma$	$(\nu^i_j[y_\alpha], \nu^i_j[x_\alpha], \nu^i_j[y_\beta], \nu^i_j[x_\beta], \nu^i_j[x_\gamma], \nu^i_j[y_\gamma])$
$\overline{v_\alpha} \vee v_\beta \vee \overline{v_\gamma}$	$(\nu^i_j[y_\alpha], \nu^i_j[x_\alpha], \nu^i_j[x_\beta], \nu^i_j[y_\beta], \nu^i_j[y_\gamma], \nu^i_j[x_\gamma])$
$v_\alpha \vee \overline{v_\beta} \vee \overline{v_\gamma}$	$(\nu^i_j[x_\alpha], \nu^i_j[y_\alpha], \nu^i_j[y_\beta], \nu^i_j[x_\beta], \nu^i_j[y_\gamma], \nu^i_j[x_\gamma])$
$\overline{v_\alpha} \vee \overline{v_\beta} \vee \overline{v_\gamma}$	$(\nu^i_j[y_\alpha], \nu^i_j[x_\alpha], \nu^i_j[y_\beta], \nu^i_j[x_\beta], \nu^i_j[y_\gamma], \nu^i_j[x_\gamma])$

For any clause in the left column, the corresponding entry in the right column above will be matched with the 6-bit string in Column A of row j of Table 1 to determine the value of ν^i_j at each bit in the support set S_i.

In Definition 3.1, we defined a correspondence between $\mathcal{M}'$ and truth assignments for Γ. In Definition 3.3, we introduced the notation $\mathcal{M}'_\Gamma(\mathcal{C}_i)$ for the collection of medians in $\mathcal{M}'$ which correspond to satisfying truth assignments for Γ. Similarly, we defined $\mathcal{M}'_{c_i}(\mathcal{C}_i)$ for each clause c_i in Γ. For the remainder of this subsection, set

$$\mathcal{M}'_\Gamma := \mathcal{M}'_\Gamma(\mathcal{C}_i), \qquad\qquad \mathcal{M}'_{c_i} := \mathcal{M}'_{c_i}(\mathcal{C}_i).$$

The following claim uses the correspondence in Definition 3.13 to connect $\mathcal{M}' \setminus \mathcal{M}'_{c_i}$ with a particular equivalence class.

Claim 3.14. *Let c_i be a clause in Γ. For any $\mu \in \mathcal{M}'$, μ is in the equivalence class represented by Column M8 of Table 1 if and only if $\mu \in \mathcal{M}' \setminus \mathcal{M}'_{c_i}$.*

Proof. Fix a clause c_i with variables $v_\alpha, v_\beta, v_\gamma$. This clause may have some negative literals. We focus our attention on v_α. The arguments for v_β and v_γ are exactly the same.

There are two cases depending on whether v_α appears as a positive literal or a negative literal in c_i.

In the case where v_α appears in c_i as a positive literal, the truth assignment which makes c_i false assigns a value of false to v_α. A corresponding median $\mu \in \mathcal{M}'$ has $\mu[x_\alpha] = 0$ and $\mu[y_\alpha] = 1$. Because v_α appears as a positive literal in c_i, the entry in the second column of Table 2 has $\mu[x_\alpha]$ followed by $\mu[y_\alpha]$. So, in this case, the 6-bit string which heads the column for medians in $\mathcal{M}' \setminus \mathcal{M}'_{c_i}$ has 01 in the first two entries.

In the case where v_α appears as a negative literal in c_i, the non-satisfying truth assignments for c_i must have v_α true. The corresponding medians $\mu \in \mathcal{M}'$ will have $\mu[x_\alpha] = 1$ and $\mu[y_\alpha] = 0$. For the clauses with variable v_α appearing as a negative literal in c_i, a quick glance at Table 2 reveals that $\mu[y_\alpha]$ immediately precedes $\mu[x_\alpha]$ in the 6-bit column headings in Table 1. As a result, the column representing medians in $\mathcal{M}' \setminus \mathcal{M}'_{c_i}$ has 01 in the first two entries.

Repeating this argument for v_β and v_γ, we see that medians in $\mathcal{M}' \setminus \mathcal{M}'_{c_i}$ are represented by the column with heading 010101. $\square$

Now that we have defined the rows and columns of Table 1, we conclude this subsection by defining the entries within Table 1 for fixed clause c_i.

Let $\mu \in \mathcal{M}'$ be an arbitrary median that falls into the equivalence class represented by Column Mℓ for some $\ell \in [8]$. The entry $a_{j\ell}$ in Row j and Column Mℓ of Table 1 is $H(\mu, \nu^i_j)$. This value can be calculated as follows:

- First, take the Hamming distance between the 6-bit string in Column A of Row j and the 6-bit string in the header of Column Mℓ. This is equal to the Hamming distance between the restrictions of μ and ν^i_j to the support set S_i for c_i.
- For any $s \notin \{\alpha, \beta, \gamma\}$, $\mu[x_s] \neq \mu[y_s]$ and $\nu^i_j[x_s] = \nu^i_j[y_s]$. Therefore the Hamming distance between $(\mu[x_s], \mu[y_s])$ and $(\nu^i_j[x_s], \nu^i_j[y_s])$ is 1 for each $s \in [n] \setminus \{\alpha, \beta, \gamma\}$.

- Finally, because $\mu[e_s] = 0$ for all $s \in [t]$, the Hamming distance between the restrictions of μ and ν_j^i to the coordinates $(e_1, e_2, \ldots, e_t)$ is the number of additional ones in ν_j^i which is found in Column C of Row j.

Adding these three values together gives the entry $a_{j\ell}$.

3.3. **Distinguishing the satisfying truth assignments.** Fix a clause c_i in arbitrary D3CNF Γ. For this subsection, we again set $\mathcal{M}' := \mathcal{M}'(\mathcal{C}_i)$, $\mathcal{M}'_\Gamma := \mathcal{M}'_\Gamma(\mathcal{C}_i)$, and $\mathcal{M}'_{c_i} := \mathcal{M}'_{c_i}(\mathcal{C}_i)$. For each $\mu \in \mathcal{M}' \setminus \mathcal{M}'_{c_i}$, μ is in the equivalence class represented by Column M8 according to Claim 3.14. Then reading the entries in Column M8 of Table 1, we find the multiset (where parenthetical subscripts give the multiplicity of that value in the multiset):

$$\begin{aligned}\{H(\mu, \nu_j^i) : j \in [50]\} = \{&(n-2)_{(1)}, (n-1)_{(6)}, n_{(3)}, (n+1)_{(15)},\\ &(n+2)_{(15)}, (n+3)_{(3)}, (n+4)_{(6)}, (n+5)_{(1)}\}.\end{aligned} \tag{1}$$

Otherwise, for each median $\mu \in \mathcal{M}'_{c_i}$, μ is in one of 7 equivalence classes represented in Columns M1 through M7. The entries in each of these columns yields

$$\{H(\mu, \nu_j^i) : j \in [50]\} = \{(n-1)_{(7)}, n_{(6)}, (n+1)_{(12)}, (n+2)_{(12)}, (n+3)_{(6)}, (n+4)_{(7)}\}. \tag{2}$$

Therefore,we can use $\mathcal{C}_i$ to distinguish between the medians in $\mathcal{M}'_{c_i}$ and the medians in $\mathcal{M}' \setminus \mathcal{M}'_{c_i}$. For example, given $\mu \in \mathcal{M}' = \{01, 10\}^n \times \{0\}^t$, if we determine that $(n+5) \in \{H(\mu, \nu_j^i) : j \in [50]\}$, then we can conclude $\mu \in \mathcal{M}' \setminus \mathcal{M}'_{c_i}$.

Now we wish to consider all of the C_i multisets together. It is clear that each x_i and each y_i coordinates will remain ambiguous in the multiset $\biguplus_{i\in[k]} \mathcal{C}_i$. For the additional ones, we will take t large enough to maintain the property that, for each $i \in [t]$, there is at most one binary string η in $\biguplus_{i\in[k]} \mathcal{C}_i$ with $\eta[e_i] = 1$. As a result,

$$\mathcal{M}\left(\biguplus_{i\in[k]} \mathcal{C}_i\right) = \{0,1\}^n \times \{0\}^t.$$

Further,

$$\mathcal{M}'_{c_i} := \mathcal{M}'_{c_i}(\mathcal{C}_i) = \mathcal{M}'_{c_i}\left(\biguplus_{i\in[k]} \mathcal{C}_i\right),$$

$$\mathcal{M}'_\Gamma := \mathcal{M}'_\Gamma(\mathcal{C}_i) = \mathcal{M}'_\Gamma\left(\biguplus_{i\in[k]} \mathcal{C}_i\right).$$

By definition of the sets $\mathcal{M}'_{c_i}$ and $\mathcal{M}'_\Gamma$,

$$\mathcal{M}'_\Gamma = \bigcap_{i\in[k]} \mathcal{M}'_{c_i}, \tag{3}$$

$$\mathcal{M}' \setminus \mathcal{M}'_\Gamma = \mathcal{M}' \setminus \bigcap_{i\in[k]} \mathcal{M}'_{c_i} = \bigcup_{i\in[k]} \left(\mathcal{M}' \setminus \mathcal{M}'_{c_i}\right). \tag{4}$$

Therefore the multiset $\biguplus_{i\in[k]} \mathcal{C}_i$ will serve as a tool to distinguish $\mathcal{M}'_\Gamma$ from $\mathcal{M}' \setminus \mathcal{M}'_\Gamma$.

4. Complexity of computing $Z(B, x!)$

Before stating Theorem 4.4, we need a result which is equivalent to the Prime Number Theorem. Define

$$\theta(x) := \sum_{\substack{p\leq x\\ p\ prime}} \log p.$$

Theorem 4.1. $\theta(x) \sim x$.

As a result, the next lemma and corollary hold.

Lemma 4.2 (Rosser 1941)**.** *For $2 \leq x$,*

$$\left(1 - \frac{2.85}{\log x}\right) x \leq \theta(x) \leq \left(1 + \frac{2.85}{\log x}\right) x.$$

Corollary 4.3. *For any $n \geq 300$,*

$$e^{n/2} \leq \prod_{\substack{p \leq n \\ p\ prime}} p \leq e^{3n/2}.$$

Now we can prove the main result for star trees.

Theorem 4.4. *Calculating $Z(B, x!)$ is #P-complete.*

Proof. In Lemma 2.13, we verified that calculating $Z(B, x!)$ is in #P . To show #P-complete, we give a polynomial time reduction from #D3SAT. Fix an arbitrary D3CNF $\Gamma = c_1 \wedge c_2 \wedge \ldots \wedge c_k$ where each c_i is a clause and Γ has n variables.

Using the bound in Corollary 4.3, let $n' = \max\{300, n+5\}$. Fix a prime number p which is greater than n' and at most $5n'$. Let

$$q := p - (n+5).$$

We will explicitly define a multiset

$$\mathcal{D}(p) = \mathcal{A}(p) \cup \bigcup_{i \in [n]} \mathcal{B}_i(p) \cup \bigcup_{i \in [k]} \mathcal{C}_i(p)$$

consisting of $2 + 2n + 50k$ binary strings with coordinates

$$(x_1, y_1, x_2, y_2, \ldots, x_n, y_n, e_1, \ldots, e_{t(p)})$$

where

$$t(p) := 2(q+4) + 2n(q+3) + k(75 + 50q). \tag{5}$$

The coordinates $e_1, e_2, \ldots, e_{t(p)}$ are for the *additional ones*. In order to define each $\eta \in \mathcal{D}(p)$, we will give exact values for $\eta[x_j]$ and $\eta[y_j]$ for each $j \in [n]$ and specify the number of additional ones that η will have. Definition 3.4 tells how to obtain the values of $\eta[e_j]$ for each $j \in [t(p)]$ from this information.

All strings in $\mathcal{D}(p)$ will come in pairs which are complementary on the first $2n$ entries (Definition 3.8). As a result, we can use Fact 3.9 to see that each of the first $2n$ coordinates are ambiguous in $\mathcal{D}(p)$.

Now we begin defining the strings in multiset that together create $\mathcal{D}(p)$. The set $\mathcal{A}(p)$ consists of two strings, α and $\overline{\alpha}$. Define α to have $\alpha[x_i] = \alpha[y_i] = 1$ for all $i \in [n]$ and $q+4$ additional ones. Define $\overline{\alpha}$ to be complementary to α on the first $2n$ entries and have $q+4$ additional ones.

For each $j \in [n]$, the set $\mathcal{B}_j(p)$ will consist of two strings, β_j and $\overline{\beta_j}$. Define β_j to be the string with $\beta_j[x_j] = \beta_j[y_j] = 1$ and for all $j' \in [n]$ with $j' \neq j$, $\beta_j[x_{j'}] = \beta_j[y_{j'}] = 0$ and $q+3$ additional ones. Define $\overline{\beta_j}$ to be complementary to β_j on the first $2n$ entries and have $q+3$ additional ones.

For each $i \in [k]$, the set $\mathcal{C}_i(p)$ will have 50 strings. These are obtained by adding q more additional ones to the 50 strings in $\mathcal{C}_i$ which were defined through Table 1 (see Section 3.1). In other words, increase each entry in Column C of Table 1 by q to obtain $\mathcal{C}_i(p)$.

In summary, we have constructed the strings

$$\mathcal{D}(p) := \mathcal{A}(p) \cup \bigcup_{i \in [n]} \mathcal{B}_i(p) \cup \bigcup_{i \in [k]} \mathcal{C}_i(p).$$

As described in Definition 3.3 and for each clause c_i in Γ, set

$$\mathcal{M}(p) := \mathcal{M}(\mathcal{D}(p)), \qquad \mathcal{M}'(p) := \mathcal{M}'(\mathcal{D}(p)),$$
$$\mathcal{M}'_{c_i}(p) := \mathcal{M}'_{c_i}(\mathcal{D}(p)), \qquad \mathcal{M}'_{\Gamma}(p) := \mathcal{M}'_{\Gamma}(\mathcal{D}(p)).$$

As stated in Fact 3.5, each $\mu \in \mathcal{M}(p)$ has $\mu[e_j] = 0$ for all $j \in [t(p)]$. Additionally, because all of the strings in $\mathcal{D}(p)$ come in complementary pairs, the coordinates x_j and y_j are ambiguous for each $j \in [n]$ (Fact 3.9). Thus there are 2^{2n} medians μ. More precisely,

$$\mathcal{M}(p) = \{0,1\}^{2n} \times \{0\}^{t(p)} \text{ and} \tag{6}$$
$$\mathcal{M}'(p) = \{01, 10\}^{n} \times \{0\}^{t(p)}.$$

Define

$$\mathcal{H}(\mu, \mathcal{A}(p)) := \prod_{a \in \mathcal{A}(p)} H(\mu, a)!$$

and likewise define $\mathcal{H}(\mu, \mathcal{B}_j(p))$ and $\mathcal{H}(\mu, \mathcal{C}_i(p))$ for each $j \in [n]$ and $i \in [k]$. Therefore the number of scenarios admitted by median μ can be expressed by

$$\mathcal{H}(\mu) := \mathcal{H}(\mu, \mathcal{A}(p)) \cdot \prod_{i \in [n]} \mathcal{H}(\mu, \mathcal{B}_i(p)) \cdot \prod_{i \in [k]} \mathcal{H}(\mu, \mathcal{C}_i(p)).$$

At this point, we wish to calculate $\mathcal{H}(\mu) \mod p$ for each median $\mu \in \mathcal{M}(p)$. To analyze $\mathcal{H}(\mu)$ for each $\mu \in \mathcal{M}$, we define the following 3 properties that a median $\mu \in \mathcal{M}(p)$ may have.

Property 1. $\sum_{i \in [n]} (\mu[x_i] + \mu[y_i]) = n$.

Property 2. $\mu \in \mathcal{M}'(p)$.

Property 3. $\mu \in \mathcal{M}'_\Gamma(p)$.

First notice that these properties are nested. Any $\mu \in \mathcal{M}(p)$ with Property 2 must also have Property 1. Likewise, if μ has Property 3, it will also have Property 2. The next 4 claims divide $\mathcal{M}(p)$ into 4 classes and examine $\mathcal{H}(\mu)$ for medians in each class.

Claim 4.5. *For arbitrary $\mu \in \mathcal{M}(p)$, if μ does not have Property 1, and consequently does not have Property 2 or 3, then $\mathcal{H}(\mu) \equiv 0 \mod p$.*

Proof. Let μ be an arbitrary median in $\mathcal{M}(p)$. For $\alpha \in \mathcal{A}(p)$, Fact 3.7 gives

$$H(\mu, \alpha) + H(\mu, \overline{\alpha}) = 2n + (q+4) + (q+4) = 2p - 2.$$

Hence, there is an integer r such that $q + 4 \leq r \leq 2n + q + 4$ and $H(\mu, \alpha) = r$ with

$$\mathcal{H}(\mu, \mathcal{A}(p)) = r!(2p-2-r)!.$$

Since μ does not have Property 1, we can conclude that exactly one of the following holds:

$$H(\mu, \alpha) \geq (n+1) + (q+4) = p, \text{ or}$$
$$H(\mu, \overline{\alpha}) \geq (n+1) + (q+4) = p.$$

Therefore, either $r \geq p$ or $(2p-2-r) \geq p$. In the first case, $r!$ is divisible by p and, in the second, $(2p-2-r)!$ is divisible by p. Therefore $\mathcal{H}(\mu, \mathcal{A}(p)) \equiv 0 \mod p$ and consequently $\mathcal{H}(\mu) \equiv 0 \mod p$. □

Claim 4.6. *For an arbitrary $\mu \in \mathcal{M}(p)$, if μ has Property 1, but does not have Property 2, then $\mathcal{H}(\mu) \equiv 0 \mod p$.*

Proof. Suppose $\mu \in \mathcal{M}(p) \setminus \mathcal{M}'(p)$ but μ has Property 1. Because $\mu \notin \mathcal{M}'(p)$, there is an integer $j_0 \in [n]$ such that $\mu[x_{j_0}] = \mu[y_{j_0}]$. In the case when $\mu[x_{j_0}] = 0$, we have $H(\mu, \beta_{j_0}) = (n+2) + (q+3) = p$. Otherwise $\mu[x_i] = 1$ which implies $H(\mu, \overline{\beta_{j_0}}) = (n+2) + (q+3) = p$. In either case,

$$\mathcal{H}(\mu, \mathcal{B}_{j_0}(p)) = p!(p-4)!$$

and consequently $\mathcal{H}(\mu) \equiv 0 \mod p$. □

Claim 4.7. *For an arbitrary $\mu \in \mathcal{M}(p)$, if μ has Properties 1 and 2, but does not have Property 3, then $\mathcal{H}(\mu) \equiv 0 \mod p$.*

Proof. Let μ be in $\mathcal{M}'(p) \setminus \mathcal{M}'_\Gamma(p)$. Since μ corresponds to a truth assignment which does not satisfy Γ, there is a clause c_{i_0} in Γ which is not satisfied by this truth assignment. Therefore $\mu \in \mathcal{M}'(p) \setminus \mathcal{M}'_{c_{i_0}}(p)$. By (1), before adding the q additional ones to each string from $\mathcal{C}_{i_0}$, we have

$$\begin{aligned}\{H(\mu, \nu_j^{i_0}) : \nu_j^{i_0} \in \mathcal{C}_{i_0}\} = \{&(n-2)_{(1)}, (n-1)_{(6)}, n_{(3)}, (n+1)_{(15)}, \\ &(n+2)_{(15)}, (n+3)_{(3)}, (n+4)_{(6)}, (n+5)_{(1)}\}.\end{aligned} \tag{7}$$

To create $\mathcal{C}_{i_0}(p)$, we added q additional ones to each string in $\mathcal{C}_{i_0}$ which increased each Hamming distance by q. Therefore

$$\begin{aligned}\{H(\mu, \nu_j^{i_0}) : \nu_j^{i_0} \in \mathcal{C}_{i_0}(p)\} = \{&(p-7)_{(1)}, (p-6)_{(6)}, (p-5)_{(3)}, (p-4)_{(15)}, \\ &(p-3)_{(15)}, (p-2)_{(3)}, (p-1)_{(6)}, p_{(1)}\}.\end{aligned}$$

As a result,

$$\mathcal{H}(\mu,\mathcal{C}_{i_0}(p)) = (p-7)!(p-6)!^6(p-5)!^3(p-4)!^{15}(p-3)!^{15}(p-2)!^3(p-1)!^6 p!$$

which is divisible by p. Therefore $\mathcal{H}(\mu) \equiv 0 \mod p$. □

Claim 4.8. *For an arbitrary $\mu \in \mathcal{M}(p)$ having Properties 1, 2, and 3, the value*

$$\mathcal{H}(\mu) = (p-6)!^{7k}(p-5)!^{6k}(p-4)!^{12k}(p-3)!^{12k}(p-2)!^{6k+2n}(p-1)!^{7k+2},$$

which is not congruent to 0 modulo p.

Proof. Let $\mu \in \mathcal{M}'_\Gamma(p)$. Because it has Property 1,

$$\mathcal{H}(\mu,\mathcal{A}(p)) = (n+(q+4))!^2 = (p-1)!^2.$$

Since μ has Property 2, for any $i \in [n]$,

$$\mathcal{H}(\mu,\mathcal{B}_i(p)) = (n+(q+3))!^2 = (p-2)!^2.$$

Finally, μ satisfies Property 3 which means $\mu \in \mathcal{M}'_{c_i}(p)$ for all clauses c_i in Γ.

Recall that each string $\eta \in \mathcal{C}_i(p)$ is created from a string $\eta' \in \mathcal{C}_i$ by adding q more additional ones. Therefore $H(\mu,\eta) = H(\mu,\eta') + q$. So, the multiset $\mathcal{H}(\mu,\mathcal{C}_i(p))$ can be obtained from $\mathcal{H}(\mu,\mathcal{C}_i)$ found in (2) by adding q to each element. As a result,

$$\mathcal{H}(\mu,\mathcal{C}_i(p)) = (p-6)!^7(p-5)!^6(p-4)!^{12}(p-3)!^{12}(p-2)!^6(p-1)!^7.$$

Therefore

$$\mathcal{H}(\mu) = (p-6)!^{7k}(p-5)!^{6k}(p-4)!^{12k}(p-3)!^{12k}(p-2)!^{6k+2n}(p-1)!^{7k+2}. \quad (8)$$

Because p is prime, $\mathcal{H}(\mu) \not\equiv 0 \mod p$. □

Set

$$T(p) := \sum_{\mu \in \mathcal{M}(p)} \mathcal{H}(\mu).$$

Set $K(p)$ equal to the function of p displayed in (8). Thus $K(p)$ is precisely the value of the number of SCJ scenarios admitted by an arbitrary $\mu \in \mathcal{M}'_\Gamma(p)$. If we calculate $T(p)$ mod p, the four claims show that

$$T(p) \equiv \sum_{\mu \in \mathcal{M}'_\Gamma(p)} \mathcal{H}(\mu) \equiv |\mathcal{M}'_\Gamma(p)| \cdot K(p) \mod p. \quad (9)$$

If γ is the number of satisfying truth assignments for Γ, then $\gamma = |\mathcal{M}'_\Gamma(p)|$ by Definition 3.3. Therefore

$$\gamma \cdot K(p) \equiv T(p) \mod p.$$

Since p does not divide $K(p)$ (Claim 4.8), there exists an integer $K'(p)$ such that $K(p) \cdot K'(p) \equiv 1 \bmod p$. Thus

$$\gamma \equiv K'(p) \cdot T(p) \mod p.$$

While this alone is not sufficient to determine the value of γ, we can repeat this construction for many different prime values to obtain more congruences.

Recall p was fixed to be a prime greater than n' and at most $5n'$. Repeat the above construction for each prime $p_1, p_2, \ldots, p_m$ in this range. The result is a list of congruences:

$$\begin{aligned}\gamma &\equiv K'(p_1) \cdot T(p_1) \mod p_1,\\ \gamma &\equiv K'(p_2) \cdot T(p_2) \mod p_2,\\ &\vdots\\ \gamma &\equiv K'(p_m) \cdot T(p_m) \mod p_m.\end{aligned}$$

Because $p_1, p_2, \ldots, p_m$ are all prime, the Chinese Remainder Theorem guarantees a solution for γ which is unique modulo $\prod_{i \in [m]} p_i$. By the Corollary 4.3,

$$\prod_{i \in [m]} p_i = \frac{\displaystyle\prod_{\substack{p \leq 5n' \\ p\ prime}} p}{\displaystyle\prod_{\substack{p \leq n' \\ p\ prime}} p} \geq \frac{e^{5n'/2}}{e^{3n'/2}} = e^{n'} \geq e^n.$$

Since γ is the number of satisfying truth assignments for Γ, and there are only n literals which can realize one of two values, $\gamma \leq 2^n$. Since $\prod_{i \in [m]} p_i \geq e^n > 2^n \geq \gamma$, the Chinese Remainder Theorem gives the exact value of γ.

In summary, for D3CNF Γ with n variables and k clauses, we use the Sieve of Eratosthenes to identify the primes between n' and $5n'$. This runs in $O(n^2)$ time. Then for each prime p in this interval (which is at most $\max\{2n, 600\}$ primes), we create $50k + 2n + 2$ binary strings of length $2n + t(p)$ where $t(p)$ is a polynomial in n and p with $p \in O(n)$. Finally, the Chinese Remainder Theorem will solve the system of congruences in $O(\log^2(p_1 p_2 \ldots p_m))$ time (Bach and Shallit 1996). For us, this is $O(n^2 \log^2 n)$ because each prime is at most $5n$ and $m \leq 2n$.

Therefore, if we had algorithm to determine the value of $Z(B, x!)$ which ran in time polynomial in the size of B and the length of the strings in B, then we have created here a polynomial time algorithm to determine the number of satisfying truth assignments for a D3CNF, a problem which is known to be #P-complete. This finishes the proof. □

5. Stochastic Approximations for $Z(B, x!)$

In the previous section, we proved calculating $Z(B, x!)$ is a #P-complete problem. The natural next question is whether or not this value can be approximated. Viewing this problem as counting most parsimonious scenarios for the star phylogenetic tree with leaves labeled by the strings in B, we are also interested in a near uniform sampler of these labelings.

Definition 5.1. *A counting problem #A in #P has an FPAUS (fully polynomial almost uniform sampler) if there is a randomized algorithm such that, for any instance of #A and any $\epsilon > 0$, the algorithm outputs an element $x \in X$, the solution space for #A, with probability $p(x)$ where*

$$\frac{1}{2} \sum_{x \in X} |p(x) - U(x)| \leq \epsilon$$

where U is the uniform distribution on X and the algorithm runs in time polynomial in the size of the instance of #A and $-\log \epsilon$.

Definition 5.2. *A counting problem #A in #P has an FPRAS (fully polynomial randomized approximation scheme) if there is a randomized algorithm such that, for any instance of #A and any $\epsilon, \delta > 0$, the algorithm outputs an approximation $\hat{f}$ for the true answer f of the counting problem satisfying the following inequality*

$$P\left(\frac{f}{1+\epsilon} \leq \hat{f} \leq f(1+\epsilon)\right) \geq 1 - \delta \qquad (10)$$

Furthermore, the algorithm runs in time polynomial in the size of the instance #A, ϵ^{-1}, and $-\log(\delta)$.

The modulo prime number computation technique which was used to prove that calculating $Z(B, x!)$ is in #P-complete has been used to show that other problems are #P-complete. For example, Brightwell and Winkler (1991) used this technique to prove that counting the number of linear extensions of a partially order set is #P-complete. For this same problem, Karzanov and Khachiyan (1991) found a rapidly mixing Markov chain to sample the linear extensions. Since counting the linear extensions of a partially ordered set is a self-reducible counting problem, this means that it also has an FPRAS (Jerrum, Valiant, and Vazirani 1986). This may suggest that our problem of counting most parismonious scenarios also has an FPAUS and FPRAS. However, here we give a straightforward Markov chain to sample the most parsimonious scenarios that turns out to be torpidly mixing, suggesting that our problem may not have an FPAUS. With evidence for both the positive answer and the negative answer, the question of whether or not there is an FPAUS and FPRAS for $Z(B, x!)$ remains open.

Recall that a median μ for $B = \{\nu_i\}_{i=1}^m$ minimizes $\sum_{i\in[m]} H(\nu_i, \mu)$. Therefore, in the k^{th} bit, the value of μ must agree with a majority of the strings in B. If exactly half of the strings in B have a 1 in the k^{th} bit, then μ may take either a 0 or a 1 in the k^{th} bit. We call such a bit an *ambiguous bit.* Therefore a median for B is determine by the value it takes in the ambiguous bits. As a result, if B has an odd number of strings, then there is exactly one median. Here we assume that the size of B is even.

Define a primer Markov chain, P, to transition between the medians. As mentioned, it suffices to define our Markov Chain on the state space of all possible values that a median could take on the ambiguous bits. From any median, make a transition with the following probabilities:

- With probability 1/2, remain in the current state.
- With probability 1/2, randomly and uniformly select an ambiguous bit and change its value.

Because we remain at the current state with probability $\frac{1}{2}$, by definition this is a lazy Markov chain.

Observation 5.3. *The primer Markov chain P is irreducible and aperiodic.*

For a fixed multiset of strings $B = \{\nu_i\}_{i=1}^m$ and median $\mu \in \mathcal{M}(B)$, define

$$f(\mu) := \prod_{i=1}^{m} H(\mu, \nu_i).$$

Now we employ the Metropolis-Hastings algorithm (Metropolis et al. 1953) to obtain a secondary Markov chain C with a desired limit distribution as follows. The states remain the same, but the transition probabilities are changed in the following way. From state μ, we propose a next state μ' which differs from μ in at most one bit. If μ' is different from μ, accept this transition with probability

$$\min\left\{1, \frac{f(\mu')}{f(\mu)}\right\}.$$

In other words, if μ' was reached from μ with probability $P(\mu'|\mu)$, then in the secondary Markov chain C the transition from μ to μ' will be made with probability

$$C(\mu'|\mu) = P(\mu'|\mu) \cdot \min\left\{1, \frac{f(\mu')}{f(\mu)}\right\}.$$

For a given collection of strings, the function f defines a probability distribution θ on the medians where $\theta(\mu)$ is directly proportional to $f(\mu)$. In other words,

$$\theta(\mu) \propto f(\mu) \tag{11}$$

or $\theta(\mu) = kf(\mu)$ for some constant k and any median μ.

Observation 5.4. *Markov chain C is reversible and converges to the limit distribution θ.*

Therefore, we have a Markov chain on the state space of medians which, in the limit, will sample each median μ with distribution proportional to $\prod_{i=1}^m H(\mu, \nu_i)$. Once we have a median, it is easy to uniformly sample from the scenarios that it admits.

Now we will show that the Markov chain C is torpidly mixing (not rapidly mixing). To prove this result, we will need the following definitions.

For any nonempty subset S of the set of medians $\mathcal{M}(B)$, the *capacity of S* is

$$\theta(S) := \sum_{\mu\in S} \theta(\mu)$$

and the *ergodic flow out of S* is

$$F(S) := \sum_{\substack{\mu\in S \\ \nu\in\mathcal{M}(B)\setminus S}} \theta(\mu)C(\mu|\nu).$$

The *conductance* is

$$\Phi := \min\left\{\frac{F(S)}{\theta(S)} : S \subseteq M, 0 < \theta(S) \le \frac{1}{2}\right\}.$$

Theorem 5.5 (Bremaud 2008)**.** *A Markov chain is rapidly mixing if and only if $\Phi \geq \frac{1}{p(n)}$ for some polynomial $p(n)$ which is not identically zero.*

Consider the following instance of $Z(B, x!)$. Define ν and $\overline{\nu}$ to be the strings in $\{0,1\}^n$ where ν is the string of all 0s and $\overline{\nu}$ is the string of all 1s. Let $B = \{\nu_i\}_{i=1}^{2t}$ be the multiset containing t copies of ν and t copies of $\overline{\nu}$. The set of medians $\mathcal{M}(B)$ is equal to $\{0,1\}^n$. Further, if μ has exactly k ones, then

$$\prod_{i=1}^{2t} H(\mu, \nu_i)! = (k!(n-k)!)^t .$$

Consequently

$$Z(B, x!) = \sum_{\mu \in \mathcal{M}(B)} \prod_{i=1}^{2t} H(\mu, \nu_i) = \sum_{k=0}^{n} \binom{n}{k} (k!(n-k)!)^t := T.$$

Therefore $\theta(\mu) = \frac{1}{T} (k!(n-k)!)^t$.

Suppose n is odd. Consider the subset S which contains all medians with at most $\lfloor \frac{n}{2} \rfloor$ ones. For this subset, the capacity is $\theta(S) = \frac{1}{2}$.

Let S' be the set of medians μ in S with exactly $\lfloor \frac{n}{2} \rfloor$ ones. Let $\hat{S}$ be the set of medians ν in $\mathcal{M} \setminus S$ with exactly $\lceil \frac{n}{2} \rceil$ ones. Then $|S'| = \binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{\lceil \frac{n}{2} \rceil} = |\hat{S}|$. For each $\mu \in S \setminus S'$ and $\nu \in \mathcal{M} \setminus S$, $C(\mu|\nu) = 0$. Further, for each $\mu \in S'$, there are only $\lceil \frac{n}{2} \rceil$ medians ν in $\mathcal{M} \setminus S$ such that $C(\mu|\nu) \neq 0$ and for these medians $\nu \in \hat{S}$ and $C(\mu|\nu) = \frac{1}{2} \cdot \frac{1}{n}$.

For the ergodic flow out of S, we have

$$\begin{aligned}
F(S) &= \sum_{\substack{\mu \in S \\ \nu \in \mathcal{M} \setminus S}} \theta(\mu) C(\mu|\nu) \\
&= \sum_{\substack{\mu \in S' \\ \nu \in \mathcal{M} \setminus S}} \frac{1}{T} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^t C(\mu|\nu) \\
&= \sum_{\mu \in S'} \frac{1}{T} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^t \frac{1}{2} \frac{1}{n} \left\lceil \frac{n}{2} \right\rceil \\
&= \binom{n}{\lfloor \frac{n}{2} \rfloor} \frac{1}{T} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^t \frac{1}{2} \frac{1}{n} \left\lceil \frac{n}{2} \right\rceil \\
&= \frac{1}{2n} \left\lceil \frac{n}{2} \right\rceil \frac{n!}{T} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^{t-1} \\
&= \frac{1}{2n} \frac{n+1}{2} \frac{n!}{n! \sum_{k=0}^{n} (k!(n-k)!)^{t-1}} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^{t-1} \\
&\leq \frac{1}{2} \frac{1}{(0!n!)^{t-1}} \left(\left\lfloor \frac{n}{2} \right\rfloor ! \left\lceil \frac{n}{2} \right\rceil ! \right)^{t-1} \\
&\leq \frac{1}{2} \frac{1}{\binom{n}{\lfloor \frac{n}{2} \rfloor}^{t-1}}.
\end{aligned}$$

This implies

$$\Phi \leq \frac{F(S)}{\pi(S)} \leq \frac{1}{\binom{n}{\lfloor \frac{n}{2} \rfloor}^{t-1}} \leq \left(\frac{n+1}{2^n} \right)^{t-1} \leq \left(\frac{2^{n/2}}{2^n} \right)^{t-1} = \frac{1}{2^{n(t-1)/2}}.$$

Therefore, if $t > 1$, then as n grows, we see that Φ cannot be lower-bounded by a function of the form $\frac{1}{p(n)}$ where p is a polynomial in n. Therefore the Markov chain C is torpidly mixing by Theorem 5.5.

6. Complexity of computing $Z(B, f(x))$

In this section, we consider the generalized $Z(B, f(x))$. First, fix a continuous function $f : \mathbb{R} \to \mathbb{R}$. Then define the following problem:

Definition 6.1. *Given an arbitrary $m \in \mathbb{Z}^+$, let $B = \{\nu_i\}_{i=1}^m$ be an arbitrary multiset of binary strings. Determine the value of*

$$\sum_{\mu \in \mathcal{M}(S)} \prod_{i \in [m]} f(H(\nu_i, \mu)).$$

In the previous section, we showed that computing $Z(B, x!)$ is #P-complete. Here we work toward determining the computational complexity of $Z(B, f(x))$ for various functions $f(x)$. First, we formalize a definition and develop a couple of tools.

Definition 6.2. *A function $g : \mathbb{R} \to \mathbb{R}$ is strictly concave up if for any $x, y, z \in \mathbb{R}$, $x < y < z$,*

$$\frac{g(z) - g(x)}{z - x} > g(y).$$

Lemma 6.3. *If $\log f(x)$ is a strictly concave up function, then for any $x < y$ and $a > 0$,*

$$\frac{f(x)f(y)}{f(x-a)f(y+a)} < 1.$$

Proof. By the intermediate value theorem, there are real values c, d with $c \in (x - a, x)$ and $d \in (y, y + a)$ such that

$$(\log f)'(c) = \frac{1}{a}(\log f(x) - \log f(x - a)) \text{ and}$$
$$(\log f)'(d) = \frac{1}{a}(\log f(y + a) - \log f(y)).$$

Because $(\log f)'(x)$ is strictly increasing, $g'(c) < g'(d)$. Therefore,

$$\frac{1}{a}(\log f(x) - \log f(x - a)) < \frac{1}{a}(\log f(y + a) - \log f(y))$$
$$\log f(x) - \log f(x - a) < \log f(y + a) - \log f(y)$$
$$\log \frac{f(x)}{f(x-a)} < \log \frac{f(y+a)}{f(y)}$$
$$\frac{f(x)}{f(x-a)} < \frac{f(y+a)}{f(y)}$$
$$\frac{f(x)f(y)}{f(x-a)f(y+a)} < 1.$$

□

Fact 6.4. *Fix $k \in \mathbb{Z}^+ \cup \{0\}$. Let $f(x)$ be a function such that $\log f(x)$ is strictly concave up. Then*

$$\min_{\substack{\alpha, \beta \in \mathbb{Z}^+ \cup \{0\} \\ a+b=k}} f(a)f(b) = f\left(\left\lfloor \frac{k}{2} \right\rfloor\right) f\left(\left\lceil \frac{k}{2} \right\rceil\right).$$

Proof. Let $x = \left\lfloor \frac{k}{2} \right\rfloor$ and $y = \left\lceil \frac{k}{2} \right\rceil$. By Lemma 6.3, $f(x - a)f(y + a) < f(x)f(y)$ which gives the desired result. □

Theorem 6.5. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- *$\log f(x)$ is strictly concave up,*
- *the function values of f can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} > 1.$$

For arbitrary $m, s \in \mathbb{Z}^+$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \dots, \nu_m\}$ be a multiset of binary strings, each of length s. Then it is #P-complete to determine how many medians μ for S have

$$\prod_{i\in[m]} f\left(H(\nu_i, \mu)\right) \le D. \tag{12}$$

Proof. Fix a function $f(x)$ with the properties listed in the theorem.

If is straightforward to see that computing $Z(B, f(x))$ is in #P. Fix an instance consisting of integers m and s, real number D, and a multiset B of binary strings of length ℓ. Let μ be a binary string of the same length as each ν_i. We can verify that μ is a median in time $O(m\ell)$. Each $H(\nu_i, \mu)$ can be computed in time $O(\ell)$. Because $H(\nu_i, \mu) \le \ell$, we can compute $f(H(\nu_i, \mu))$ in time polynomial in the size of the input by the conditions on f. Finally, checking if the product is at most D is also a polynomial time calculation. Therefore computing $Z(B, f(x))$ is in #P.

To prove #P-hardness, we will provide a reduction from #D3SAT. Fix Γ, a D3CNF with n variables and k clauses, set

$$\begin{aligned}\kappa =&[f(n)]^{9k}[f(n+1)]^{22k+2kn}[f(n+2)]^{48k+12kn}\\ &\cdot [f(n+3)]^{48k+12kn}[f(n+4)]^{22k+2kn}[f(n+5)]^{9k}\end{aligned}$$

The idea is to define a multiset, $\mathcal{D}$, of binary strings with the following properties:

- Each median μ which corresponds to a satisfying truth assignment for Γ will have
$$\prod_{\eta\in\mathcal{D}} f(H(\eta, \mu)) = \kappa$$
- Each other median μ' will have
$$\prod_{\eta\in\mathcal{D}} f(H(\eta, \mu')) > \kappa.$$

Create a total of $158k + 28kn$ strings of length $2n + 260k + 35kn$ with coordinates

$$(x_1, y_1, x_2, y_2, \dots, x_n, y_n, e_1, e_2, \dots, e_t)$$

where $t = 260k + 35kn$. This multiset of binary strings will be defined as the union of three multisets:

$$\mathcal{D} = \mathcal{A} \uplus \biguplus_{i\in[n]} \mathcal{B}_i \uplus \biguplus_{i\in[k]} \mathcal{C}_i'.$$

As in Definition 3.4, we will define each string $\eta \in \mathcal{D}$ by explicitly giving the values of $\eta[x_i]$ and $\eta[y_i]$ for each $i \in [n]$ and telling the number of additional ones.

The collection $\mathcal{A}$ contains $108k$ strings. For $a \in [t]$, let $\alpha^{(+a)}$ be the string with $\alpha[x_i] = \alpha[y_i] = 1$ for all $1 \le i \le n$ and a additional ones. Define $\overline{\alpha}^{(+a)}$ to be the binary string which is complementary to $\alpha^{(+0)}$ on the first $2n$ coordinates and has a additional ones. The multiset $\mathcal{A}$ will consist of the following strings:

- k copies each of $\alpha^{(+0)}$ and $\overline{\alpha}^{(+0)}$,
- 8k copies each of $\alpha^{(+1)}$ and $\overline{\alpha}^{(+1)}$,
- 18k copies each of $\alpha^{(+2)}$ and $\overline{\alpha}^{(+2)}$,
- 18k copies each of $\alpha^{(+3)}$ and $\overline{\alpha}^{(+3)}$,
- 8k copies each of $\alpha^{(+4)}$ and $\overline{\alpha}^{(+4)}$,
- k copies each of $\alpha^{(+5)}$ and $\overline{\alpha}^{(+5)}$.

The collection $\mathcal{B} = \biguplus_{i\in[n]} \mathcal{B}_i$ contains $28kn$ strings. For each $i \in [n]$, $a \in [t]$, let $\beta_i^{(+a)}$ be the string with $\beta_i[x_i] = \beta_i[y_i] = 1$, $\beta_i[x_j] = \beta_i[y_j] = 0$ for $j \ne i$, and with a additional ones. Define the binary string $\overline{\beta_i}^{(+a)}$ to be complementary to $\beta_i^{(+0)}$ on the first $2n$ coordinates and have a additional ones. The collection $\mathcal{B}_i$ consists of the following $28k$ strings:

- k copies each of $\beta_i^{(+1)}$ and $\overline{\beta_i}^{(+1)}$,
- $6k$ copies each of $\beta_i^{(+2)}$ and $\overline{\beta_i}^{(+2)}$,
- $6k$ copies each of $\beta_i^{(+3)}$ and $\overline{\beta_i}^{(+3)}$,
- k copies each of $\beta_i^{(+4)}$ and $\overline{\beta_i}^{(+4)}$.

The collection $\mathcal{C} = \biguplus_{i\in[k]} \mathcal{C}'_i$ contains $50k$ strings. Each set $\mathcal{C}'_i$, which is associated with clause c_i, consists of 50 strings. In Section 3.1, we defined set $\mathcal{C}_i$ through Table 1. For each $\nu^i_j \in \mathcal{C}_i$, create $\hat{\nu}^i_j$ by increasing the number of additional ones in ν^i_j by one. Then

$$\mathcal{C}'_i := \{\hat{\nu}^i_j : \nu^i_j \in \mathcal{C}_i\}.$$

Let $\mathcal{M}$ be the set of all medians for $\mathcal{D}$. From Definition 3.3 and for each clause c_i in Γ, set

$$\mathcal{M} := \mathcal{M}(\mathcal{D}), \qquad \mathcal{M}' := \mathcal{M}'(\mathcal{D}),$$
$$\mathcal{M}'_{c_i} := \mathcal{M}'_{c_i}(\mathcal{D}), \qquad \mathcal{M}'_\Gamma := \mathcal{M}'_\Gamma(\mathcal{D}).$$

According to Remark 3.4, all medians μ must have $\mu[e_i] = 0$ for all $i \in [t]$. In $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, the strings come in pairs where one is complementary to the other on the first $2n$ coordinates. By Fact 3.9, each of the x_i and y_i coordinates are ambiguous. Therefore

$$\mathcal{M} = \{0,1\}^{2n} \times \{0\}^t,$$
$$\mathcal{M}' = \{01,10\}^{n} \times \{0\}^t.$$

Define

$$\mathcal{H}(\mu,\mathcal{A}) := \prod_{a\in\mathcal{A}} f(H(\mu,a)).$$

Similarly define $\mathcal{H}(\mu,\mathcal{B}_i)$ and $\mathcal{H}(\mu,\mathcal{C}'_j)$. Set

$$\mathcal{H}(\mu) := \mathcal{H}(\mu,\mathcal{A}) \cdot \prod_{i\in[n]} \mathcal{H}(\mu,\mathcal{B}_i) \cdot \prod_{j\in[k]} \mathcal{H}(\mu,\mathcal{C}'_j).$$

For each $\mu \in \mathcal{M}$, we obtain a lower bound for $\mathcal{H}(\mu)$ and for each $\mu \in \mathcal{M}'_\Gamma$ we describe an exact value for $\mathcal{H}(\mu)$. Divide $\mathcal{M}$ into 4 classes using the following three properties which a median $\mu \in \mathcal{M}$ may have.

Property 1. $\sum_{i\in[n]}(\mu[x_i] + \mu[y_i]) = n$.
Property 2. $\mu \in \mathcal{M}'$.
Property 3. $\mu \in \mathcal{M}'_\Gamma$.

Notice that these properties are nested. Any median $\mu \in \mathcal{M}$ with Property 2, must also have Property 1. Further, any $\mu \in \mathcal{M}$ with Property 3 must also have Property 2. The following claims provide lower bounds for medians according to their properties.

Claim 6.6. *If $\mu \in \mathcal{M}$ has Property 1, then*

$$\begin{aligned}\mathcal{H}(\mu,\mathcal{A}) =&[f(n)]^{2k}[f(n+1)]^{16k}[f(n+2)]^{36k}\\ &\cdot [f(n+3)]^{36k}[f(n+4)]^{16k}[f(n+5)]^{2k}\\ =:&\alpha_{good}.\end{aligned} \tag{13}$$

Otherwise,

$$\begin{aligned}\mathcal{H}(\mu,\mathcal{A}) \geq&[f(n-1)f(n+1)]^{k}[f(n)f(n+2)]^{8k}[f(n+1)f(n+3)]^{18k}\\ &\cdot [f(n+2)f(n+4)]^{18k}[f(n+3)f(n+5)]^{8k}[f(n+4)f(n+6)]^{k}\\ =:&\alpha_{bad}.\end{aligned} \tag{14}$$

Proof. If $\mu \in \mathcal{M}$ has Property 1, then $H(\mu,\alpha^{(+0)}) = H(\mu,\overline{\alpha}^{(+0)}) = n$ because $\alpha^{(+0)}[x_i] = \alpha^{(+0)}[y_i] = 1$ for all $i \in [n]$ while μ only has n ones in the first 2n entries. Because $\mu[e_i] = 0$ for all $i \in [t]$, by Definition 3.4,

$$H(\mu,\alpha^{(+a)}) = H(\mu,\overline{\alpha}^{(+a)}) = n + a.$$

Recalling the exact strings that appear in $\mathcal{A}$, we quickly obtain (13).

If μ does not have Property 1, then either μ has more than n ones in the first $2n$ entries, implying $H(\mu,\alpha^{(+0)}) > n$, or μ has less than n ones in the first $2n$ entries, implying $H(\mu,\overline{\alpha}^{(+0)}) > n$. By Fact 3.7,

$H(\mu,\alpha^{(+0)}) + H(\mu,\overline{\alpha}^{(+0)}) = 2n$. By Fact 6.4 and the above observations,

$$f\left(H\left(\mu,\alpha^{(+0)}\right)\right)\cdot f\left(H(\mu,\overline{\alpha}^{(+0)})\right) \geq f(n-1)f(n+1),$$
$$f\left(H\left(\mu,\alpha^{(+a)}\right)\right)\cdot f\left(H\left(\mu,\overline{\alpha}^{(+a)}\right)\right) \geq f(n-1+a)f(n+1+a).$$

Recalling the exact strings in $\mathcal{B}$, we obtain the lower bound in (14). □

Claim 6.7. *For each $\mu \in \mathcal{M}$ and each $i \in [n]$,*

$$\mathcal{H}(\mu,\mathcal{B}_i) \geq [f(n+1)]^{2k}[f(n+2)]^{12k}[f(n+3)]^{12k}[f(n+4)]^{2k} =: \beta_{good}. \tag{15}$$

If μ has Property 2, then for every $i \in [n]$,

$$\mathcal{H}(\mu,\mathcal{B}_i) =: \beta_{good}.$$

If μ satisfies Property 1, but not Property 2, then there exists $i_0 \in [n]$ such that

$$\begin{aligned}\mathcal{H}(\mu,\mathcal{B}_{i_0}) =&[f(n-1)f(n+3)]^k[f(n)f(n+4)]^{6k}\\ &\cdot[f(n+1)f(n+5)]^{6k}[f(n+2)f(n+6)]^k\\ =:&\beta_{bad}.\end{aligned} \tag{16}$$

Proof. For any $\mu \in \mathcal{M}$, by Fact 3.7,

$$H(\mu,\beta^{(+0)}) + H(\mu,\overline{\beta}^{(+0)}) = 2n.$$

By Fact 6.4, for each $a \in \mathbb{Z}^+ \cup \{0\}$,

$$f\left(H\left(\mu,\beta^{(+0)}\right)\right)\cdot f\left(H\left(\mu,\overline{\beta}^{(+0)}\right)\right) \geq [f(n)]^2, \text{ and}$$
$$f\left(H\left(\mu,\beta^{(+a)}\right)\right)\cdot f\left(H\left(\mu,\overline{\beta}^{(+a)}\right)\right) \geq [f(n+a)]^2.$$

Therefore, for any $\mu \in \mathcal{M}$,

$$\mathcal{H}(\mu,\mathcal{B}_i) \geq \beta_{good}.$$

If $\mu \in \mathcal{M}'$, then for each $i \in [n]$, $\mu[x_i] \neq \mu[y_i]$. On the other hand, for each $i \in [n]$, $j \in [n]$, $\beta_i^{(+a)}[x_j] = \beta_i^{(+a)}[y_j]$. Therefore for any $i,j \in [n]$,

$$H((\mu[x_j],\mu[y_j]),(\beta_i^{(+a)}[x_j],\beta_i^{(+a)}[y_j])) = 1.$$

The same holds if $\beta_i^{(+a)}$ is replaced with $\overline{\beta}_i^{(+a)}$. Therefore,

$$H\left(\mu,\beta_i^{(+0)}\right) = H\left(\mu,\overline{\beta}_i^{(+0)}\right) = n,$$
$$H\left(\mu,\beta_i^{(+a)}\right) = H\left(\mu,\overline{\beta}_i^{(+a)}\right) = n+a.$$

As a result $\mathcal{H}(\mu) = \beta_{good}$.

If μ satisfies Property 1 but not Property 2, then we can define a tighter lower bound on $\mathcal{H}(\mu,\mathcal{B}_i)$. In particular, because $\mu \notin \mathcal{M}'$, there exists $i_0 \in [n]$ such that $\mu[x_{i_0}] = \mu[y_{i_0}]$. Recall $\beta_{i_0}^{(+a)}[x_{i_0}] = \beta_{i_0}^{(+a)}[y_{i_0}] = 1$ and $\overline{\beta}_{i_0}^{(+a)}[x_{i_0}] = \overline{\beta}_{i_0}^{(+a)}[y_{i_0}] = 0$. Therefore,

$$\begin{aligned}\mu[x_{i_0}] = 1 \Rightarrow &H((\mu[x_{i_0}],\mu[y_{i_0}]),(\beta_{i_0}^{(+a)}[x_{i_0}],\beta_{i_0}^{(+a)}[y_{i_0}])) = 0,\\ &H((\mu[x_{i_0}],\mu[y_{i_0}]),(\overline{\beta}_{i_0}^{(+a)}[x_{i_0}],\overline{\beta}_{i_0}^{(+a)}[y_{i_0}])) = 2, \text{ and}\\ \mu[x_{i_0}] = 0 \Rightarrow &H((\mu[x_{i_0}],\mu[y_{i_0}]),(\overline{\beta}_{i_0}^{(+a)}[x_{i_0}],\overline{\beta}_{i_0}^{(+a)}[y_{i_0}])) = 0,\\ &H((\mu[x_{i_0}],\mu[y_{i_0}]),(\beta_{i_0}^{(+a)}[x_{i_0}],\beta_{i_0}^{(+a)}[y_{i_0}])) = 2.\end{aligned}$$

Because μ satisfies Property 1, there are exactly n ones among the first $2n$ coordinates. Without loss of generality, $\mu[x_{i_0}] = \mu[y_{i_0}] = 1$. Set

$$S := \{x_j, y_j : j \in [n], j \neq i_0\}.$$

Then μ has $n-2$ ones and n zeros among the coordinates in S. However, $\beta_{i_0}^{(+a)}$ takes the value 0 on each of the coordinates of S and $\overline{\beta}_{i_0}^{(+a)}$ takes the value 1 on the coordinates of S. Therefore,

$$\begin{aligned}\mu[x_{i_0}] = 1 \Rightarrow & H(\mu, \beta_{i_0}^{(+0)}) = 0 + (n-2),\\ & H(\mu, \overline{\beta}_{i_0}^{(+0)}) = 2 + n, \text{ and}\\ \mu[x_{i_0}] = 0 \Rightarrow & H(\mu, \overline{\beta}_{i_0}^{(+0)}) = 0 + (n-2),\\ & H(\mu, \beta_{i_0}^{(+0)}) = 2 + n.\end{aligned}$$

As a result,

$$\begin{aligned}H(\mu, \beta_{i_0}^{(+0)})H(\mu, \overline{\beta}_{i_0}^{(+0)}) &= (n-2)(n+2),\\ H(\mu, \beta_{i_0}^{(+a)})H(\mu, \overline{\beta}_{i_0}^{(+a)}) &= (n-2+a)(n+2+a).\end{aligned}$$

Taking into account all binary strings in $\mathcal{B}_{i_0}$, we conclude $\mathcal{H}(\mu, \mathcal{B}_{i_0}) = \beta_{bad}$ in (16). ☐

Fact 6.8. *For the quantities defined in Claim 6.7, $\beta_{good} < \beta_{bad}$. Consequently, if $\mu \in \mathcal{M} \setminus \mathcal{M}'$ and satisfies Property 1, then $\prod_{i\in[n]} \mathcal{H}(\mu, \mathcal{B}_i) \geq \beta_{bad}\beta_{good}^{k-1}$. If $\mu \in \mathcal{M} \setminus \mathcal{M}'$ and does not satisfy Property 1, then $\prod_{i\in[n]} \mathcal{H}(\mu, \mathcal{B}_i) \geq \beta_{good}^{k}$.*

Proof. Observe

$$\begin{aligned}\frac{\beta_{bad}}{\beta_{good}} &= \left[\frac{f(n-1)f(n)^6 f(n+1)^4 f(n+4)^4 f(n+5)^6 f(n+6)}{f(n+2)^{11} f(n+3)^{11}}\right]^k\\ &= \left[\frac{f(n-1)f(n+6)}{f(n+2)f(n+3)}\right]^k \left[\frac{f(n)f(n+5)}{f(n+2)f(n+3)}\right]^{6k} \left[\frac{f(n+1)f(n+4)}{f(n+2)f(n+3)}\right]^{4k}\\ &> 1\end{aligned}$$

where the last inequality follows from Lemma 6.3. ☐

Claim 6.9. *For any $\mu \in \mathcal{M}$ and for each $j \in [k]$,*

$$\mathcal{H}(\mu, \mathcal{C}_i') \geq [f(n+2)]^{25}[f(n+3)]^{25} =: \gamma_{min}.$$

If $\mu \in \mathcal{M}_\Gamma'$, then for each $j \in [k]$,

$$\mathcal{H}(\mu, \mathcal{C}_i') = [f(n)]^7[f(n+1)]^6[f(n+2)]^{12}[f(n+3)]^{12}[f(n+4)]^6[f(n+5)]^7 =: \gamma_{good}. \tag{17}$$

If $\mu \in \mathcal{M}' \setminus \mathcal{M}_\Gamma'$, then there exists $i_0 \in [k]$ such that

$$\begin{aligned}\mathcal{H}(\mu, \mathcal{C}_{i_0}') =& f(n-1)[f(n)]^6[f(n+1)]^3[f(n+2)]^{15}\\ & \cdot [f(n+3)]^{15}[f(n+4)]^3[f(n+5)]^6 f(n+6)\\ =& : \gamma_{bad}.\end{aligned} \tag{18}$$

Proof. Let μ be an arbitrary median in $\mathcal{M}$. By Remark 3.12, the binary strings in $\mathcal{C}_i$ come in pairs that are complementary on the first $2n$ entries. With a careful examination of Table 1, if $\eta, \eta' \in \mathcal{C}_i$ are complementary on the first $2n$ coordinates, then $e(\eta) + e(\eta') = 3$ where e is the function specifying the number of additional ones. By the definition of $\mathcal{C}_i'$, the strings still come in complementary pairs, $(\hat{\eta}, \hat{\eta}')$, but here $e(\hat{\eta}) + e(\hat{\eta}') = 5$ because the number of additional ones in $\hat{\eta}$ and $\hat{\eta}'$ is precisely one more than the number in η and η'. By Fact 3.7, for each of the 25 pairs in $\mathcal{C}_i'$,

$$H(\mu, \hat{\eta}) + H(\mu, \hat{\eta}') = 2n + 5.$$

Then by Fact 6.4,

$$f(H(\mu, \hat{\eta}))f(H(\mu, \hat{\eta}')) \geq f(n+2)f(n+3)$$

which gives the general bound γ_{min}.

Now suppose $\mu \in \mathcal{M}'_\Gamma$. This implies $\mu \in \mathcal{M}'_{c_i}$ for all clauses c_i in Γ. By the definition of $\mathcal{C}'_i$, for each $\hat{\nu}^i_j \in \mathcal{C}'_i$, $H(\mu, \hat{\nu}^i_j) = H(\mu, \nu^i_j) + 1$ where $\nu^i_j \in \mathcal{C}_i$. From (2), we see

$$\{H(\mu, \hat{\nu}^i_j) : j \in [50]\} = \{n_{(7)}, (n+1)_{(6)}, (n+2)_{(12)}, (n+3)_{(12)}, (n+4)_{(6)}, (n+5)_{(7)}\}.$$

This immediately implies $\mathcal{H}(\mu, \mathcal{C}'_i) = \gamma_{good}$ in (17).

Finally, suppose $\mu \in \mathcal{M}' \setminus \mathcal{M}'_\Gamma$. Using the bijection in Definition 3.1, μ must correspond to a truth assignment which does not satisfy Γ. So there is a clause c_{i_0} in Γ which is not satisfied. Therefore $\mu \in \mathcal{M}' \setminus \mathcal{M}'_{c_{i_0}}$. From (1), adding 1 to each $H(\mu, \nu^{i_0}_j)$ to obtain $H(\mu, \hat{\nu}^{i_0}_j)$, we obtain

$$\begin{aligned}\{H(\mu, \nu^{i_0}_j) : j \in [50]\} = \{&(n-1)_{(1)}, n_{(6)}, (n+1)_{(3)}, (n+2)_{(15)},\\ &(n+3)_{(15)}, (n+4)_{(3)}, (n+5)_{(6)}, (n+6)_{(1)}\}.\end{aligned} \tag{19}$$

This directly implies $\mathcal{H}(\mu, \mathcal{C}_{i_0}) = \gamma_{bad}$ in (18). □

Fact 6.10. *For the quantities defined in Claim 6.9, $\gamma_{good} < \gamma_{bad}$. As a result, when $\mu \in \mathcal{M}' \setminus \mathcal{M}'_\Gamma$,*

$$\mathcal{H}(\mu, \mathcal{C}) \geq \gamma_{bad}\gamma_{good}^{k-1}.$$

Proof. Indeed, this was our initial assumption:

$$\frac{\gamma_{bad}}{\gamma_{good}} = \frac{f(n-1)[f(n+2)]^3[f(n+3)]^3 f(n+6)}{f(n)[f(n+1)]^3[f(n+4)]^3[f(n+5)]} > 1.$$

The bound for $\mathcal{H}(\mu, \mathcal{C})$ results from the fact that $\mu \in \mathcal{M}'$ either corresponds to a satisfying truth assignment for c_i or a non-satisfying truth assignment for each clause c_i. □

In summary, Claims 6.6, 6.7, and 6.9 along with Facts 6.8 and 6.10, we give the following bounds. If $\mu \in \mathcal{M}'_\Gamma$,

$$\mathcal{H}(\mu) = \alpha_{good}\beta^n_{good}\gamma^k_{good} =: h_3.$$

If $\mu \in \mathcal{M}' \setminus \mathcal{M}'_\Gamma$,

$$\mathcal{H}(\mu) \geq \alpha_{good}\beta^n_{good}\gamma_{bad}\gamma^{k-1}_{good} =: h_2.$$

If $\mu \in \mathcal{M} \setminus \mathcal{M}'$ and has Property 1,

$$\mathcal{H}(\mu) \geq \alpha_{good}\beta_{bad}\beta^{n-1}_{good}\gamma^k_{min} =: h_1.$$

If $\mu \in \mathcal{M}$ but does not have Property 1,

$$\mathcal{H}(\mu) \geq \alpha_{bad}\beta^n_{good}\gamma^k_{min} =: h_0.$$

In order to complete, the proof, we only need to show $h_3 < h_i$ for $i \in \{0, 1, 2\}$. By one of our assumptions about $f(x)$, we have already verified in Fact 6.10 that

$$\frac{h_2}{h_3} = \frac{\gamma_{bad}}{\gamma_{good}} > 1.$$

Next observe

$$\begin{aligned}\frac{h_1}{h_2} =&\frac{\beta_{bad}}{\beta_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}\gamma_{good}^{k-1}}\\ >&\frac{\beta_{bad}}{\beta_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}^k}\\ =&\left[\frac{f(n-1)f(n+3)}{[f(n+1)]^2}\left[\frac{f(n)f(n+4)}{[f(n+2)]^2}\right]^6\left[\frac{f(n+1)f(n+5)}{[f(n+3)]^2}\right]^6\frac{f(n+2)f(n+6)}{[f(n+4)]^2}\right]^k\\ &\cdot\left[\frac{[f(n+2)]^{10}[f(n+3)]^{10}}{f(n-1)[f(n)]^6[f(n+1)]^3[f(n+4)]^3[f(n+5)]^6f(n+6)}\right]^k\\ =&\left[\frac{f(n+1)f(n+4)}{f(n+2)f(n+3)}\right]^k\\ >&1\end{aligned}$$

where the last inequality follows from Lemma 6.3.

Finally we prove that $h_0 > h_2$.

$$\begin{aligned}\frac{h_0}{h_2} =&\frac{\alpha_{bad}}{\alpha_{good}}\frac{\gamma_{min}^k}{\gamma_{bad}\gamma_{good}^{k-1}}\\ >&\frac{\alpha_{bad}}{\alpha_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}^k}\\ =&\left[\frac{f(n-1)f(n+1)}{[f(n)]^2}\left[\frac{f(n)f(n+2)}{[f(n+1)]^2}\right]^8\left[\frac{f(n+1)f(n+3)}{[f(n+2)]^2}\right]^{18}\right.\\ &\cdot\left.\left[\frac{f(n+2)f(n+4)}{[f(n+3)]^2}\right]^{18}\left[\frac{f(n+3)f(n+5)}{[f(n+4)]^2}\right]^8\frac{f(n+4)f(n+6)}{[f(n+5)]^2}\right]^k\\ &\cdot\left[\frac{[f(n+2)]^{10}[f(n+3)]^{10}}{f(n-1)[f(n)]^6[f(n+1)]^3[f(n+4)]^3[f(n+5)]^6f(n+6)}\right]^k\\ =&\frac{[f(n-1)]^k[f(n)]^{8k}[f(n+1)]^{19k}[f(n+2)]^{26k}}{[f(n)]^{2k}[f(n+1)]^{16k}[f(n+2)]^{36k}}\\ &\cdot\frac{[f(n+3)]^{26k}[f(n+4)]^{19k}[f(n+5)]^{8k}[f(n+6)]^k}{[f(n+3)]^{36k}[f(n+4)]^{16k}[f(n+5)]^{2k}}\\ &\cdot\frac{[f(n+2)]^{10k}[f(n+3)]^{10k}}{[f(n-1)]^k[f(n)]^{6k}[f(n+1)]^{3k}[f(n+4)]^{3k}[f(n+5)]^{6k}f(n+6)^k}\\ =&1.\end{aligned}$$

Therefore for any $\hat{\mu} \in \mathcal{M}'_\Gamma$ and $\mu \in \mathcal{M} \setminus \mathcal{M}'_\Gamma$, then $\mathcal{H}(\hat{\mu}) < \mathcal{H}(\mu)$. Thus, if we could determine, in polynomial time, how many medians $\mu \in \mathcal{M}$ have $h(\mu) \leq h_3$, then we could determine how many satisfying truth assignments exist for Γ in polynomial time. □

Corollary 6.11. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- *$\log f(x)$ is strictly concave up,*
- *the function values of f can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3f(n+4)} > 1.$$

For arbitrary $m, s \in \mathbb{Z}^+$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \dots, \nu_m\}$ be a multiset of binary strings, each of length s and let $\mathcal{M}$ be the set of medians for S. Then it is NP-complete to determine if

$$\min_{\mu\in\Gamma} \prod_{i\in[m]} f\left(H(\nu_i,\mu)\right) \leq D. \tag{20}$$

This next theorem gives the same result as Theorem 6.5 with one change in the conditions on f. While Theorem 6.5 required that

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} > 1,$$

Theorem 6.12 switches the inequality to consider functions in which the ratio is less than 1.

Theorem 6.12. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- $\log f(x)$ *is strictly concave up,*
- *the function values of f can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} < 1.$$

For arbitrary $m, s \in \mathbb{N}$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \dots, \nu_m\}$ be a multiset of binary strings, each of length s. Then it is #P-complete to determine how many medians μ for S have

$$\prod_{i\in[m]} f\left(H(\nu_i,\mu)\right) \leq D.$$

Proof. This proof closely mirrors the proof of Theorem 6.5. Here we will note the changes that need to be made.

This time, we define $98k + 24kn$ binary strings, each of length $2n + 245k + 60kn$ with coordinates

$$(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t).$$

Let $\alpha^{(+a)}$ and $\overline{\alpha}^{(+a)}$ be defined as before. The collection $\mathcal{A}$ will now consist of the following $72k$ strings:

- $4k$ copies each of $\alpha^{(+1)}$ and $\overline{\alpha}^{(+1)}$,
- $14k$ copies each of $\alpha^{(+2)}$ and $\overline{\alpha}^{(+2)}$,
- $14k$ copies each of $\alpha^{(+3)}$ and $\overline{\alpha}^{(+3)}$,
- $4k$ copies each of $\alpha^{(+4)}$ and $\overline{\alpha}^{(+4)}$.

Define $\beta_i^{(+a)}$ and $\overline{\beta}_i^{(+a)}$ as before. The collection $\mathcal{B}_i$ now consists of the following $24k$ strings:

- $6k$ copies each of $\beta_i^{(+2)}$ and $\overline{\beta}_i^{(+2)}$,
- $6k$ copies each of $\beta_i^{(+3)}$ and $\overline{\beta}_i^{(+3)}$.

Following the explanation found in Section 3.1, Table 6 defines 26 binary strings $\overline{\mathcal{C}_i}$ for a clause. As in the proof of Theorem 6.5, we will add 1 additional one to each of the 26 strings in $\overline{\mathcal{C}_i}$ to create $\mathcal{C}_i'$.

Using the same Properties 1, 2, and 3 as before, we obtain the following values which are analogous to the bounds in Claims 6.6, 6.7, and 6.9:

$$\begin{aligned}
\alpha_{good} :=& [f(n+1)]^{8k}[f(n+2)]^{28k}[f(n+3)]^{28k}[f(n+4)]^{8k},\\
\alpha_{bad} :=& [f(n)f(n+2)]^{4k}[f(n+1)f(n+3)]^{14k}\\
&\cdot [f(n+2)f(n+4)]^{14k}[f(n+3)f(n+5)]^{4k},
\end{aligned}$$

$$\begin{aligned}
\beta_{good} :=& [f(n+2)]^{12k}[f(n+3)]^{12k},\\
\beta_{min} :=& [f(n+2)]^{12k}[f(n+3)]^{12k},\\
\beta_{bad} :=& [f(n)f(n+4)]^{6k}[f(n+1)f(n+5)]^{6k},\\
\gamma_{good} :=& f(n-1)[f(n)]^3[f(n+1)]^3[f(n+2)]^6\\
&\cdot [f(n+3)]^6[f(n+4)]^3[f(n+5)]^3 f(n+6),\\
\gamma_{bad} :=& [f(n)]^4[f(n+1)]^6[f(n+2)]^3[f(n+3)]^3[f(n+4)]^6[f(n+5)]^4,\\
\gamma_{min} :=& [f(n+2)]^{13}[f(n+3)]^{13}.
\end{aligned}$$

By our assumption about $f(x)$,

$$\frac{\gamma_{bad}}{\gamma_{good}} = \frac{f(n)[f(n+1)]^3[f(n+4)]^3 f(n+5)}{f(n-1)[f(n+2)]^3[f(n+3)]^3 f(n+6)} > 1$$

which implies $\gamma_{bad} > \gamma_{good}$.

Next we determine the values of h_0, h_1, h_2, h_3 in this setting. As before, if μ has Property i, but not property $i+1$, then $\mathcal{H}(\mu) \geq h_i$. Further, if μ has Property 3, $\mathcal{H}(\mu) = h_3$. If μ does not have Property 1, then $\mathcal{H}(\mu) \geq h_0$.

$$\begin{aligned}
h_3 &:= \alpha_{good}\beta_{good}^n\gamma_{good}^k.\\
h_2 &:= \alpha_{good}\beta_{good}^n\gamma_{bad}\gamma_{good}^{k-1}.\\
h_1 &:= \alpha_{good}\beta_{bad}\beta_{min}^{n-1}\gamma_{min}^k.\\
h_0 &:= \alpha_{bad}\beta_{min}^n\gamma_{min}^k.
\end{aligned}$$

As in the proof of Theorem 6.5, we will show $h_3 < h_0, h_1, h_2$.

By our assumption about $f(x)$,

$$\frac{h_2}{h_3} = \frac{\gamma_{bad}}{\gamma_{good}} > 1.$$

Also

$$\begin{aligned}
\frac{h_1}{h_2} =& \frac{\beta_{bad}}{\beta_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}\gamma_{good}^{k-1}}\\
>& \frac{\beta_{bad}}{\beta_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}^k}\\
=& \left[\left[\frac{f(n)f(n+4)}{[f(n+2)]^2}\right]^6 \left[\frac{f(n+1)f(n+5)}{[f(n+3)]^2}\right]^6\right]^k\\
&\cdot \left[\frac{[f(n+2)]^{10}[f(n+3)]^{10}}{[f(n)]^4[f(n+1)]^6[f(n+4)]^6[f(n+5)]^4}\right]^k\\
=& \left[\frac{f(n)f(n+5)}{f(n+2)f(n+3)}\right]^{2k}\\
>& 1 \qquad\qquad \text{by Lemma 6.3.}
\end{aligned}$$

Finally we show $h_0 > h_2$.

$$\begin{aligned}
\frac{h_0}{h_2} =& \frac{\alpha_{bad}}{\alpha_{good}} \frac{\gamma_{min}^k}{\gamma_{bad}\gamma_{good}^{k-1}} \\
>& \frac{\alpha_{bad}}{\alpha_{good}} \cdot \frac{\gamma_{min}^k}{\gamma_{bad}^k} \\
=& \left[\left[\frac{f(n)f(n+2)}{[f(n+1)]^2} \right]^4 \left[\frac{f(n+1)f(n+3)}{[f(n+2)]^2} \right]^{14} \right. \\
& \left. \cdot \left[\frac{f(n+2)f(n+4)}{[f(n+3)]^2} \right]^{14} \left[\frac{f(n+3)f(n+5)}{[f(n+4)]^2} \right]^4 \right]^k \\
& \cdot \left[\frac{[f(n+2)]^{10}[f(n+3)]^{10}}{[f(n)]^4[f(n+1)]^6[f(n+4)]^6[f(n+5)]^4} \right]^k \\
=& \frac{[f(n)]^{4k}[f(n+1)]^{14k}[f(n+2)]^{18k}[f(n+3)]^{18k}[f(n+4)]^{14k}[f(n+5)]^{4k}}{[f(n+1)]^{8k}[f(n+2)]^{28k}[f(n+3)]^{28k}[f(n+4)]^{8k}} \\
& \cdot \frac{[f(n+2)]^{10k}[f(n+3)]^{10k}}{[f(n)]^{4k}[f(n+1)]^{6k}[f(n+4)]^{6k}[f(n+5)]^{4k}} \\
=& 1.
\end{aligned}$$

Making each of these changes in the proof of Theorem 6.5, we complete the proof of Theorem 6.12. □

Table 3. The 26 strings to complement the collection in Table 1 along with their Hamming distance from medians in $\mathcal{M}'$.

	A	**B**	**C**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Row #	Values of ν_j^i on its support set	$\nu_j^i[x_\ell]$, $\nu_j^i[y_\ell]$ $(v_\ell \notin c_i)$	Add'l Ones	10 10 10	10 10 01	10 01 10	01 10 10	10 01 01	01 10 01	01 01 10	01 01 01
1	01 00 00	0	+0	$n+1$	$n+1$	$n+1$	$n-1$	$n+1$	$n-1$	$n-1$	$n-1$
2	00 01 00	0	+0	$n+1$	$n+1$	$n-1$	$n+1$	$n-1$	$n+1$	$n-1$	$n-1$
3	00 00 01	0	+0	$n+1$	$n-1$	$n+1$	$n+1$	$n-1$	$n-1$	$n+1$	$n-1$
4	10 11 11	1	+3	$n+2$	$n+2$	$n+2$	$n+4$	$n+2$	$n+4$	$n+4$	$n+4$
5	11 10 11	1	+3	$n+2$	$n+2$	$n+4$	$n+2$	$n+4$	$n+2$	$n+4$	$n+4$
6	11 11 10	1	+3	$n+2$	$n+4$	$n+2$	$n+2$	$n+4$	$n+4$	$n+2$	$n+4$
7	01 01 00	0	+2	$n+4$	$n+4$	$n+2$	$n+2$	$n+2$	$n+2$	n	n
8	01 00 01	0	+2	$n+4$	$n+2$	$n+4$	$n+2$	$n+2$	n	$n+2$	n
9	00 01 01	0	+2	$n+4$	$n+2$	$n+2$	$n+4$	n	$n+2$	$n+2$	n
10	10 10 11	1	+1	$n-1$	$n-1$	$n+1$	$n+1$	$n+1$	$n+1$	$n+3$	$n+3$
11	10 11 10	1	+1	$n-1$	$n+1$	$n-1$	$n+1$	$n+1$	$n+3$	$n+1$	$n+3$
12	11 10 10	1	+1	$n-1$	$n+1$	$n+1$	$n-1$	$n+3$	$n+1$	$n+1$	$n+3$
13	10 01 01	0	+1	$n+2$	n	n	$n+4$	$n-2$	$n+2$	$n+2$	n
14	01 10 01	0	+1	$n+2$	n	$n+4$	n	$n+2$	$n-2$	$n+2$	n
15	01 01 10	0	+1	$n+2$	$n+4$	n	n	$n+2$	$n+2$	$n-2$	n
16	10 10 01	0	+1	n	$n-2$	$n+2$	$n+2$	n	n	$n+4$	$n+2$
17	10 01 10	0	+1	n	$n+2$	$n-2$	$n+2$	n	$n+4$	n	$n+2$
18	01 10 10	0	+1	n	$n+2$	$n+2$	$n-2$	$n+4$	n	n	$n+2$
19	10 10 10	0	+1	$n-2$	n	n	n	$n+2$	$n+2$	$n+2$	$n+4$
20	01 01 01	1	+2	$n+5$	$n+3$	$n+3$	$n+3$	$n+1$	$n+1$	$n+1$	$n-1$
21	10 01 01	1	+2	$n+3$	$n+1$	$n+1$	$n+5$	$n-1$	$n+3$	$n+3$	$n+1$
22	01 10 01	1	+2	$n+3$	$n+1$	$n+5$	$n+1$	$n+3$	$n-1$	$n+3$	$n+1$
23	01 01 10	1	+2	$n+3$	$n+5$	$n+1$	$n+1$	$n+3$	$n+3$	$n-1$	$n+1$
24	10 10 01	1	+2	$n+1$	$n-1$	$n+3$	$n+3$	$n+1$	$n+1$	$n+5$	$n+3$
25	10 01 10	1	+2	$n+1$	$n+3$	$n-1$	$n+3$	$n+1$	$n+5$	$n+1$	$n+3$
26	01 10 10	1	+2	$n+1$	$n+3$	$n+3$	$n-1$	$n+5$	$n+1$	$n+1$	$n+3$

The information in this table is to be read in the same way as the information in Table 1. This is detailed in Section 3.1.

Corollary 6.13. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- $\log f(x)$ *is strictly concave up,*
- *the function values of f can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} < 1.$$

For arbitrary $m, s \in \mathbb{Z}^+$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \ldots, \nu_m\}$ be a multiset of binary strings, each of length s and let $\mathcal{M}$ be the set of medians for S. Then it is NP-complete to determine if

$$\min_{\mu \in \mathcal{M}} \prod_{i \in [m]} f\left(H(\nu_i, \mu)\right) \leq D. \tag{21}$$

We can also state the following corollaries for functions which are strictly concave down.

Corollary 6.14. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- $\log f(x)$ *is strictly concave down,*
- *the function values can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} \neq 1.$$

For arbitrary $m, s \in \mathbb{Z}^+$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \ldots, \nu_m\}$ be a multiset of binary strings, each of length s. Then it is #P-complete to determine if how many medians μ for S have

$$\prod_{i \in [m]} f\left(H(\nu_i, \mu)\right) \geq D. \tag{22}$$

Proof. If the function $f(x)$ has the property that $\log f(x)$ is strictly concave down, then $\log \frac{1}{f(x)}$ is strictly concave up. Therefore by Theorems 6.5 and 6.12 for the function $\frac{1}{f(x)}$, it is #P-hard to determine the number of medians μ which satisfy $\prod_{i \in [m]} \frac{1}{f(H(\nu_i, \mu))} \leq \frac{1}{D}$. This is equivalent to asking for the number of medians μ have $\prod_{i \in [m]} f(H(\nu_i, \mu)) \geq D$. □

Corollary 6.15. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- $\log f(x)$ *is strictly concave down,*
- *the function values of f can be computed in polynomial time, and*
- *for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} \neq 1.$$

For arbitrary $m, s \in \mathbb{Z}^+$ and $D \in \mathbb{R}$, let $S := \{\nu_1, \nu_2, \ldots, \nu_m\}$ be a multiset of binary strings, each of length s where $\mathcal{M}$ is the set of medians for S. Then it is NP-complete to determine if

$$\min_{\mu \in \mathcal{M}} \prod_{i \in [m]} f\left(H(\nu_i, \mu)\right) \geq D. \tag{23}$$

7. Stochastic Approximations for $Z(B, f(x))$

We have seen several proofs showing that it is hard to calculate many of these quantities. As in Section 5, we may further ask if any of these quantities can be approximated. We will again focus on approximations via an FPRAS (Definition 5.2).

Before stating our results, we define a couple more complexity classes for decision problems:

Definition 7.1 (Gill 1977). *A decision problem, A, is in the class RP (randomized polynomial time) if there is a probabilistic Turing machine that runs in polynomial time in the size of the input, returns "true" with probability at least $\frac{1}{2}$ when the answer for A is true, and returns "false" with probability 1 when the answer for A is false.*

Definition 7.2 (Gill 1977). *A decision problem, A, is in the class BPP (bounded-error probabilistic polynomial time) if there is a probabilistic Turing machine that runs in polynomial time in the size of the input, returns "true" with probability at least $\frac{2}{3}$ when the answer for A is true, and returns "false" with probability $\frac{2}{3}$ when the answer for A is false.*

One result connecting these classes is the following:

Theorem 7.3 (Papadimitriou 1994). *If the intersection of NP and BPP is non-empty, then RP=NP.*

Note that each result below holds for functions $f(x)$ with $\log f(x)$ strictly concave down. The analogous results for the functions whose logarithm is concave up are still open. Our first result can be interpreted as sampling medians for #SPSCJ with a probability distribution analogous to the number of scenarios, but dependent on $f(x)$.

Theorem 7.4. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ which satisfies the following properties:*

- *$\log f(x)$ is strictly concave down,*
- *the function values of f can be computed in polynomial time, and*
- *there exists $\epsilon > 0$ such that for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} < 1 - \epsilon.$$

For arbitrary $m, s \in \mathbb{Z}^+$, let $S := \{\nu_1, \nu_2, \ldots, \nu_m\}$ be a multiset of binary strings, each of length s. If there is a rapidly mixing Markov chain with stationary distribution proportional to $\prod_{i\in[m]} f(H(\nu_i, \mu))$, then RP=NP.

Proof. Fix a function f as described in the theorem. Because $\log f(x)$ is strictly concave down, $\log(f(x))^{-1}$ is strictly concave up. Set $g(x) := (f(x))^{-1}$.

Now recall the proof of Theorem 6.5 for strictly concave up functions. Take a D3CNF Γ with n variables and create a multiset of binary strings, $\mathcal{D}$. The set of medians for $\mathcal{D}$ is $\mathcal{M} = \{0,1\}^{2n} \times \{0\}^t$. There is a one-to-one correspondence between the medians in the subset $\mathcal{M}' = \{01, 10\}^n \times \{0\}^t$ and the truth assignments for Γ. Those medians which correspond to satisfying truth assignments for Γ form the set $\mathcal{M}'_\Gamma$. The multiset $\mathcal{D}$ is constructed so that each $\mu \in \mathcal{M}'_\Gamma$ has

$$\prod_{\eta\in\mathcal{D}} \frac{1}{f(H(\eta,\mu))} = \prod_{\eta\in\mathcal{D}} g(H(\eta,\mu)) = \alpha_{good}\beta_{good}^n\gamma_{good}^k =: h_3$$

and all other medians have

$$\prod_{\eta\in\mathcal{D}} \frac{1}{f(H(\eta,\mu))} = \prod_{\eta\in\mathcal{D}} g(H(\eta,\mu)) > \alpha_{good}\beta_{good}^n\gamma_{bad}\gamma_{good}^{k-1} =: h_2.$$

Equivalently, if $\mu \in \mathcal{M}'_\Gamma$, then

$$\prod_{\eta\in\mathcal{D}} f(H(\eta,\mu)) = \frac{1}{h_3}.$$

Otherwise,

$$\prod_{\eta\in\mathcal{D}} f(H(\eta,\mu)) < \frac{1}{h_2}.$$

Further,

$$\begin{aligned}
\frac{h_2}{h_3} &= \frac{\gamma_{bad}}{\gamma_{good}} \\
&= \frac{g(n-1)[g(n+2)]^3[g(n+3)]^3 g(n+6)}{g(n)[g(n+1)]^3[g(n+4)]^3[g(n+5)]} \\
&= \frac{f(n)[f(n+1)]^3[f(n+4)]^3[f(n+5)]}{f(n-1)[f(n+2)]^3[f(n+3)]^3 f(n+6)} \\
&> \frac{1}{1-\epsilon}
\end{aligned}$$

where the last inequality is a result of the assumption in the theorem statement. As a result

$$\frac{1}{h_2}\left(\frac{1}{1-\epsilon}\right) < \frac{1}{h_3}.$$

Now select an integer r, dependent only on the values of n and ϵ, such that $\left(\frac{1}{1-\epsilon}\right)^r > 2^{2n+2}$. Create a new multiset $\mathcal{D}(r)$ of binary strings such that

$$\mathcal{D}(r) = \underbrace{\mathcal{D} \uplus \ldots \uplus \mathcal{D}}_{r \text{ times}}.$$

The set of medians for $\mathcal{D}(r)$ is the same as the set of medians for $\mathcal{D}$. However this time, for a median $\mu \in \mathcal{M}'_\Gamma$,

$$\prod_{\eta\in\mathcal{D}(r)} f(H(\eta,\mu)) = \left(\frac{1}{h_3}\right)^r.$$

Otherwise, if $\mu \in \mathcal{M} \setminus \mathcal{M}'_\Gamma$,

$$\prod_{\eta\in\mathcal{D}(r)} f(H(\eta,\mu)) < \left(\frac{1}{h_2}\right)^r.$$

By the choice of r,

$$\left(\frac{1}{h_2}\right)^r 2^{2n} < \left(\frac{1}{h_2}\right)^r 2^{2n+2} < \left(\frac{1}{h_2}\left(\frac{1}{1-\epsilon}\right)\right)^r < \left(\frac{1}{h_3}\right)^r.$$

Since $\mathcal{M} = \{0,1\}^{2n} \times \{0\}^t$, $|\mathcal{M}| = 2^{2n}$ and the above inequality shows that for each $\mu_0 \in \mathcal{M}'_\Gamma$,

$$\prod_{\eta\in\mathcal{D}} f(H(\eta,\mu_0)) > \sum_{\mu\in\mathcal{M}\setminus\mathcal{M}'_\Gamma} \prod_{\eta\in\mathcal{D}} f(H(\eta,\mu)).$$

Further,

$$\prod_{\eta\in\mathcal{D}} f(H(\eta,\mu_0)) > \frac{1}{2}\sum_{\mu\in\mathcal{M}} \prod_{\eta\in\mathcal{D}} f(H(\eta,\mu)).$$

Now suppose that we had a rapidly mixing Markov chain on the medians for this instance as stated in the theorem. From the calculations above, it must sample medians which correspond to satisfying truth assignments for Γ with probability at least $\frac{1}{2}$. This is precisely an RP for D3SAT. However, this immediately implies RP=NP because D3SAT is NP-complete. □

The following theorem gives the same result as the last one for different functions f. In particular, it switches the inequality that f is required to satisfy.

Theorem 7.5. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0,\infty)$ which satisfies the following properties:*

- $\log f(x)$ *is strictly concave down,*
- *the function values of f can be computed in polynomial time, and*
- *there exists $\epsilon > 0$ such that for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*

$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} > 1+\epsilon.$$

For arbitrary $m, s \in \mathbb{Z}^+$, let $S := \{\nu_1, \nu_2, \ldots, \nu_m\}$ be a multiset of binary strings, each of length s. If there is a rapidly mixing Markov chain with distribution proportional to $\prod_{i\in[m]} f(H(\nu_i,\mu))$, then RP=NP.

Proof. The proof for this theorem follows the same line of reasoning as the proof for Theorem 7.4. However, it makes use of details in Theorem 6.12 rather than Theorem 6.5. □

When f is a function with $\log f(x)$ is concave down, we examine the possibility of an FPRAS (Definition 5.2) for $Z(B, f(x))$.

Theorem 7.6. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0,\infty)$ for which:*

- $\log f(x)$ *is strictly concave down,*
- *the function values of f can be computed in polynomial time, and*

- *there exists $\epsilon > 0$ such that for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*
$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} < 1-\epsilon.$$

For arbitrary $m, s \in \mathbb{Z}^+$, let $S := \{\nu_1, \nu_2, \dots, \nu_m\}$ be a multiset of binary strings, each of length s. If there is an FPRAS for calculating
$$Z(B, f(x)) = \sum_{\mu \in \mathcal{M}} \prod_{i \in [m]} f(H(\nu_i, \mu)),$$
then RP=NP.

Proof. Let r be an integer so that $\left(\frac{1}{1-\epsilon}\right)^r > 2^{2n+2}$. In the proof of Theorem 7.4, we created a new multiset of strings $\mathcal{D}(r)$. The set of medians $\mathcal{M}$ for $\mathcal{D}(r)$ is precisely $\{0,1\}^{2n} \times \{0\}^t$ and each median $\mu \in \mathcal{M}'_\Gamma$ which corresponds to a satisfying truth assignment for Γ has
$$\prod_{\eta \in \mathcal{D}(r)} f(H(\eta, \mu)) = \left(\frac{1}{h_3}\right)^r.$$
All other medians have
$$\prod_{\eta \in \mathcal{D}(r)} f(H(\eta, \mu)) < \left(\frac{1}{h_2}\right)^r.$$
Therefore, if Γ has no satisfying assignments,
$$\sum_{\mu \in \mathcal{M}} \prod_{\eta \in \mathcal{D}(r)} f(H(\eta, \mu)) < 2^{2n}\left(\frac{1}{h_2}\right)^r.$$
If there is a satisfying assignment for Γ, then
$$\sum_{\mu \in \mathcal{M}} \prod_{\eta \in \mathcal{D}(r)} f(H(\eta, \mu)) \geq \left(\frac{1}{h_3}\right)^r.$$
By the choice of r, we have the following inequality to relate the two quantities:
$$\left(\frac{1}{h_2}\right)^r 2^{2n+2} < \left(\frac{1}{h_3}\right)^r.$$

Now suppose that there is an FPRAS for $T := \sum_{\mu \in \mathcal{M}} \prod_{\eta \in \mathcal{D}(r)} f(H(\eta, \mu))$. In other words, for any $\epsilon, \delta > 0$, there is a randomized algorithm as described in Definition 5.2 which outputs a quantity $\hat{T}$ such that
$$P\left(\frac{T}{1+\epsilon} \leq \hat{T} \leq T(1+\epsilon)\right) \geq 1-\delta.$$
Consider the case when $\delta = \frac{1}{3}$ and $\epsilon = 1$. Therefore,
$$P\left(\frac{1}{2}T \leq \hat{T} \leq 2T\right) \geq \frac{2}{3}.$$
Therefore, if Γ can be satisfied, then $T \geq \left(\frac{1}{h_3}\right)^r$ and the probability that $\hat{T}$ is at least $\frac{1}{2}T = \frac{1}{2}\left(\frac{1}{h_3}\right)^r > 2^{2n+1}\left(\frac{1}{h_2}\right)^r$ is $\frac{2}{3}$. On the other hand, if Γ cannot be satisfied, then $T < 2^{2n}\left(\frac{1}{h_2}\right)^r$ and the probability that $\hat{T}$ is at most $2T = 2^{2n+1}\left(\frac{1}{h_2}\right)^r$ is $\frac{2}{3}$. Therefore, we have a BPP algorithm (Definition 7.2) for D3SAT. Because D3SAT is NP-complete, Papadimitriou's Theorem 7.3 implies RP=NP. □

A similar result holds for functions $f(x)$ which satisfy the opposite inequality. We do not give a proof as it follows the same reasoning in the proof of Theorem 7.6.

Theorem 7.7. *Fix a function $f(x) : \mathbb{Z}^+ \cup \{0\} \to [0, \infty)$ for which:*
- *$\log f(x)$ is strictly concave down,*
- *the function values of f can be computed in polynomial time, and*

- *there exists $\epsilon > 0$ such that for all but finitely many $n \in \mathbb{Z}$, $n \geq 2$,*
$$\frac{f(n-2)[f(n+1)]^3[f(n+2)]^3 f(n+5)}{f(n-1)[f(n)]^3[f(n+3)]^3 f(n+4)} > 1+\epsilon.$$

For arbitrary $m, s \in \mathbb{Z}^+$, let $S := \{\nu_1, \nu_2, \dots, \nu_m\}$ be a multiset of binary strings, each of length s. If there is an FPRAS for calculating
$$Z(B, f(x)) = \sum_{\mu \in \mathcal{M}} \prod_{i \in [m]} f(H(\nu_i, \mu)),$$
then RP=NP.

8. Set-up for results on binary trees

Previously, we explored the value $Z(B, x!)$ as it related to the number of ways to label a star phylogenetic tree. In this section, we divert our exploration to binary phylogenetic trees. First we give a precise definition of a binary tree.

Definition 8.1. *A tree is a binary tree if it is rooted and every non-leaf vertex has exactly two children.*

Fix a multiset B of m binary strings from $\{0,1\}^n$. Also fix a binary tree T with m leaves. Label the leaves of T with the strings from B via the surjective function $\varphi : L(T) \to B$. The equivalent of a median in this setting is a labeling $\varphi' : V(T) \to \{0,1\}^n$ which agrees with φ on $L(T)$ and minimizes $\sum_{uv \in E(T)} H(\varphi'(u), \varphi'(v))$. Such a vertex labeling φ' is called a *most parsimonious labeling.* Let $\mathcal{M}(T, \varphi)$ be the set of most parsimonious labelings which extend φ for the binary tree T.

As with the star trees, we will label each edge of T with a scenario. Given a most parsimonious labeling φ' for $V(T)$, a scenario for the edge uv is a permutation of the bits in which $\varphi'(u)$ and $\varphi'(v)$ differ.

A *most parsimonious scenario* for a binary tree T with leaf labeling $\varphi : L(T) \to B$ consists of a most parsimonious labeling φ' of the vertices of T and a scenario to label each edge of T. We desire to count the number of most parsimonious scenarios which is
$$Z_{T,\varphi}(B, x!) := \sum_{\varphi' \in \mathcal{M}(T,\varphi)} \prod_{uv \in E(T)} H(\varphi'(u), \varphi'(v))!.$$

Formally, the problem statement is below:

Definition 8.2 (#Binary)**.** *Given arbitrary integer $m \geq 2$, let T be a binary tree with m leaves. Let $B = \{\nu_i\}_{i=1}^m$ be an arbitrary multiset of binary strings and let $\varphi : L(T) \to B$ be a surjective function. Determine the value of $Z_{T,\varphi}(B, x!)$.*

The main result of Section 9 is the theorem which states #Binary is in #P-complete. In this section, we develop several tools and algorithms which lay the foundation for our main theorem.

Let $\Gamma = c_1 \wedge c_2 \wedge \ldots \wedge c_k$ be a D3CNF with variables $\{v_1, v_2, \dots, v_n\}$. Select new variables $\{w_1, w_2, \dots, w_n\}$ which do not occur in Γ. For each $i \in [n]$, interpretting subscript $n+1$ as 1, define the following D3CNF,
$$\Phi_i := (v_i \vee w_i \vee v_{i+1}) \wedge (v_i \vee w_i \vee \overline{v_{i+1}}) \wedge (\overline{v_i} \vee \overline{w_i} \vee v_{i+1}) \wedge (\overline{v_i} \vee \overline{w_i} \vee \overline{v_{i+1}}). \tag{24}$$
Observe that Φ_i is equivalent to the "exclusive or" $(v_i \vee w_i) \wedge (\overline{v_i} \vee \overline{w_i})$. Define
$$\Psi(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{n} \Phi_i. \tag{25}$$
Necessarily, if Γ is a D3CNF then so is $\Psi(\Gamma)$.

Lemma 8.3. *For Γ, an arbitrary D3CNF, it is #P-complete to determine the number of satisfying truth assignments for $\Psi(\Gamma)$.*

Proof. We have already shown in Lemma 2.9 that #D3SAT is in #P-complete. So to prove this result, we will show that the satisfying truth assignments for Γ and for $\Psi(\Gamma)$ are in one-to-one correspondence.

Any truth assignment which satisfies $\Psi(\Gamma)$, when restricted to $\{v_1, v_2, \dots, v_n\}$ will necessarily satisfy Γ.

For the other direction, recall that Φ_i is equivalent to the "exclusive or" for v_i and w_i. Therefore, given a satisfying truth assignment for Γ, we can create a unique satisfying truth assignment for $\Psi(\Gamma)$ by assigning to each w_i the opposite value of v_i. $\square$

Next we provide two different algorithms for creating most parsimonious labelings given a rooted binary tree and a leaf-labeling $\varphi : L(T) \to \{0,1\}^n$. If we restrict $\varphi(\ell)$ to a single coordinate c for every leaf ℓ, we obtain a labeling $\varphi_c : L(T) \to \{0,1\}$. Each algorithm presented below will consider leaf labels from the set $\{0,1\}$ and output a most parsimonious labeling $\varphi'_c : V(T) \to \{0,1\}$. Obtaining a most parsimonious labeling for each coordinate in this way, we combine these labelings to create a most parsimonious labeling $\varphi' : V(T) \to \{0,1\}^n$ for T and the original leaf-labeling φ.

Let T be a binary tree with root ρ and let $\varphi : L(T) \to \{0,1\}$ be a labeling for the leaves. Let $\varphi' : V(T) \to \{0,1\}$ be a most parsimonious labeling which extends φ. Because each vertex is labeled with a single bit, $H(\varphi'(u), \varphi'(v)) \in \{0,1\}$ for any edge uv. By definition, the most parsimonious labeling φ' minimizes the sum $\sum_{uv \in E(T)} H(\varphi'(u), \varphi'(v))$. Consequently, φ' must minimize the number of edges uv such that $\varphi'(u) \neq \varphi'(v)$.

First, we have Fitch's algorithm to find most parsimonious labelings.

Fitch's Algorithm (Fitch 1971). *Let T be a binary tree with root ρ and leaf-labeling $\varphi : L(T) \to \{0,1\}$. The following algorithm, completed in two parts, will find a most parsimonious labeling $\varphi' : V(T) \to \{0,1\}$ which extends φ.*

Part 1: Define a function B on the vertices of T as follows: For each leaf ℓ, set $B(\ell) := \{\varphi(\ell)\}$. Extend this assignment to all vertices of T by the following rule: For a vertex u with children v_1, v_2 such that $B(v_1)$ and $B(v_2)$ have been defined, set

$$B(u) := \begin{cases} B(v_1) \cap B(v_2) & \text{if } B(v_1) \cap B(v_2) \neq \emptyset, \\ B(v_1) \cup B(v_2) & \text{otherwise.} \end{cases} \tag{26}$$

Part 2: Select a single element $\alpha \in B(\rho)$. Define a function φ' on the vertices of T as follows: Set $\varphi'(\rho) := \alpha$. Extend φ' to $V(T)$ by the following rule: If v is a child of u and $\varphi'(u)$ is defined, then

$$\varphi'(v) := \begin{cases} \varphi'(u) & \text{if } \varphi'(u) \in B(v), \\ 1 - \varphi'(u) & \text{if } \varphi'(u) \notin B(v). \end{cases} \tag{27}$$

The resulting φ' is a most parsimonious labeling extending φ and is called a Fitch solution.

While Fitch solutions are most parsimonious labelings, there are cases when Fitch's algorithm finds some of the most parsimonious labelings but not all of them. However, Sankoff's algorithm, described below, will produce all most parsimonious labelings (Erdős and Székely 1994).

Sankoff's Algorithm (Erdős and Székely 1994; Sankoff and Rousseau 1975). *Let T be a binary tree with root ρ and leaf labeling $\varphi : L(T) \to \{0,1\}$. This algorithm is completed in two steps.*

Part 1: Define functions s_0 and s_1 on the vertices of T as follows: First, for each leaf ℓ,

$$s_0(\ell) := \begin{cases} 0 & \text{if } \varphi(\ell) = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{28}$$

$$s_1(\ell) := \begin{cases} 0 & \text{if } \varphi(\ell) = 1, \\ \infty & \text{otherwise.} \end{cases}$$

Extend these functions recursively to all vertices by the following: If v_0 and v_1 are children of u and $s_i(v_j)$ has been defined for all $i, j \in \{0,1\}$, then

$$s_0(u) := \min\{s_0(v_0), s_1(v_0) + 1\} + \min\{s_0(v_1), s_1(v_1) + 1\}, \tag{29}$$

$$s_1(u) := \min\{s_0(v_0) + 1, s_1(v_0)\} + \min\{s_0(v_1) + 1, s_1(v_1)\}. \tag{30}$$

Note: For any $v \in V(T)$, $s_i(v)$ counts the minimum number of edges, within the subtree containing v and its descendants, that will witness a change if a most parsimonious labeling assigned label i to vertex v. A leaf will have $s_0(\ell) = \infty$ (or $s_1(\ell) = \infty$ if it is impossible for a most parsimonious labeling to label ℓ with a 0 (1), because most parsimonious labelings must agree with the original leaf label.

Part 2: For each $v \in V(T)$, select $\alpha_v \in \{0,1\}$. Define the function φ' on the vertices of T as follows: For root ρ, define

$$\varphi'(\rho) := \begin{cases} 0 & \text{if } s_0(\rho) < s_1(\rho), \\ \alpha_\rho & \text{if } s_0(\rho) = s_1(\rho), \\ 1 & \text{if } s_0(\rho) > s_1(\rho). \end{cases}$$

Extend φ' to $V(T)$ by the following rule: If v is a child of u and $\varphi'(u)$ is defined, then define $\varphi'(v)$ as follows: If $\varphi'(u) = 0$, then

$$\varphi'(v) := \begin{cases} 0 & \text{if } s_0(v) < s_1(v) + 1, \\ \alpha_v & \text{if } s_0(v) = s_1(v) + 1, \\ 1 & \text{if } s_0(v) > s_1(v) + 1. \end{cases} \tag{31}$$

If $\varphi'(u) = 1$, then

$$\varphi'(v) := \begin{cases} 1 & \text{if } s_1(v) < s_0(v) + 1, \\ \alpha_v & \text{if } s_1(v) = s_0(v) + 1, \\ 0 & \text{if } s_1(v) > s_0(v) + 1. \end{cases} \tag{32}$$

The resulting φ' is a most parsimonious labeling for T extending φ and is called a Sankoff solution.

The following lemma draws a connection between the solutions found from each algorithm.

Lemma 8.4. *Let T be a binary tree with leaf-labeling $\varphi : L(T) \to \{0,1\}$. Suppose that, for each $u, v \in V(T)$ with v a child of u, the function B in Fitch's algorithm satisfies*

$$B(v) = \{0,1\} \Rightarrow B(u) = \{0,1\}. \tag{33}$$

Then for T and φ, all Sankoff solutions are Fitch solutions. In other words, Fitch's algorithm finds all most parsimonious labelings.

In order to prove Lemma 8.4, we first establish a series of claims (8.5 through 8.8) under the assumptions of Lemma 8.4.

Claim 8.5. *For any non-leaf vertex v, if $B(v) = \{x\}$ for some $x \in \{0,1\}$ in Fitch's algorithm, then $s_0(v) = 0$ and $s_1(v) = 2$ in Sankoff's algorithm.*

Proof. The proof proceeds by reverse induction on the distance from the root. For the base case, we consider those vertices whose children are both leaves. Let v be such a vertex with children v_ℓ and v_r. By symmetry of the argument, assume $B(v) = \{0\}$. Then $B(v_\ell) = B(v_r) = \{0\}$ which only happens for leaves if $\varphi(v_\ell) = \varphi(v_r) = 0$. By (28), $s_0(v_\ell) = s_0(v_r) = 0$ and $s_1(v_\ell) = s_1(v_r) = \infty$. As desired, (29) implies $s_0(v) = 0$ and (30) implies $s_1(v) = 2$.

For the inductive hypothesis, assume that each vertex v of distance at least $d \geq 1$ from the root has either $s_0(v) = 0$ and $s_1(v) = 2$ or $s_0(v) = 2$ and $s_1(v) = 0$. Let u be a vertex of distance $d-1$ from the root. Again, we assume $B(u) = \{0\}$ as the argument for the case when $B(u) = \{1\}$ is very similar. This vertex has two children, u_ℓ and u_r. There are three cases to consider.

(1) If u_ℓ and u_r are leaves, then the argument in the base case gives $s_0(u) = 0$ and $s_1(u) = 2$ as desired.
(2) If u_ℓ is a leaf and u_r is not a leaf, then $s_0(u_r) = 0$ and $s_1(u_r) = \infty$ and, by the inductive hypothesis $s_0(u_\ell) = 0$ and $s_1(u_\ell) = 2$. Therefore (29) implies $s_0(u) = 0$ and (30) implies $s_1(u) = 2$.
(3) If u_ℓ and u_r are not leaves, then by the inductive hypothesis, $s_0(u_\ell) = s_0(u_r) = 0$ and $s_1(u_\ell) = s_1(u_r) = 2$. Again, (29) implies $s_0(u) = 0$ and (30) implies $s_1(u) = 2$.

This complete the proof of the claim. □

Claim 8.6. *For any vertex v with $B(v) = \{0,1\}$ from Fitch's algorithm, we will have $s_0(v) = s_1(v)$ in Sankoff's algorithm.*

Proof. This claim is also proven by induction on distance from the root where the base case examines those vertices with greatest distance from the root.

For the base case, let v be a vertex with $B(v) = \{0,1\}$ and none of its descendants u have $B(u) = \{0,1\}$. For children v_ℓ and v_r of v we may assume $B(v_\ell) = \{0\}$ and $B(v_r) = \{1\}$ by (26). By Claim 8.5,

$$s_0(v_\ell) = s_1(v_r) = 0 \text{ and } s_0(v_r) = s_1(v_\ell) = 2.$$

Therefore $s_0(v) = s_1(v) = 1$.

For the inductive hypothesis, suppose all vertices u with $B(u) = \{0,1\}$ of distance at least $d \geq 1$ from the root have $s_0(u_\ell) = s_1(u_r)$. Let v be a vertex at distance $d-1$ from the root with $B(v) = \{0,1\}$. There are three cases to consider:

(1) If v has a child v_ℓ with $B(v_\ell) = \{0\}$, then by (26) the other child v_r must have $B(v_r) = \{1\}$ and we can use the argument in the base case to see $s_0(v_\ell) = s_1(v_r)$.
(2) If v has a child v_ℓ with $B(v_\ell) = \{1\}$, then by (26), v must have another child v_r with $B(v_r) = \{0\}$. This puts us back in case 1.
(3) If v has a child v_ℓ with $B(v_\ell) = \{0,1\}$, then by (26), v must have another child v_r with $B(v_r) = \{0,1\}$. By the inductive hypothesis, $s_0(v_\ell) = s_1(v_\ell)$ and $s_0(v_r) = s_1(v_r)$. By (29) and (30), $s_0(v) = s_1(v)$.

This completes the proof of the claim. □

Claim 8.7. *For any non-leaf vertex v with $B(v) = \{i\}$ ($i \in \{0,1\}$), both Fitch's algorithm and Sankoff's algorithm will define $\varphi'(v) = i$.*

Proof. In Fitch's algorithm, this is an immediate consequence of (27).

Now consider Sankoff's algorithm. If $B(v) = \{0\}$ then, by Claim 8.5, $s_0(v) = 0$ and $s_1(v) = 2$. Observe $s_0(v) < s_1(v) + 1$ and $s_1(v) > s_0(v) + 1$. Therefore $\varphi'(v) = 0$ by (31) and (32). □

Claim 8.8. *Suppose $B(\rho) = \{0,1\}$. For any vertex v with $B(v) = \{0,1\}$, if both algorithms set $\varphi'(\rho) := 0$, then both Fitch's algorithm and Sankoff's algorithm will set $\varphi'(v) = 0$. Likewise, if $\varphi'(\rho) = 1$, then both algorithms will set $\varphi'(v) = 1$.*

Proof. For any vertex v with $B(v) = \{0,1\}$, there is a path $\rho = u_0, u_1, \dots, u_{t-1}, u_t = v$ of vertices such that $B(u_i) = \{0,1\}$ for each $i \in [t]$. It suffices to show that, in both algorithms, if $\varphi'(u_i) = 0$ for some $0 \leq i < t$, then $\varphi'(u_{i+1}) = 0$.

In Part 2 of Fitch's algorithm, if $\varphi'(u_i) = 0$ and $B(u_{i+1}) = \{0,1\}$, then (26) implies $\varphi'(u_{i+1}) = 0$.

In Sankoff's algorithm, if $\varphi'(u_i) = 0$ and $B(u_{i+1}) = \{0,1\}$, then by Claim 8.8, $s_0(u_{i+1}) = s_1(u_{i+1})$. Thus $s_0(u_{i+1}) < s_1(u_{i+1}) + 1$. Since $\varphi'(u_i) = 0$, (31) implies $\varphi'(u_{i+1}) = 0$.

A similar argument can be used to show that if $\varphi'(\rho) = 1$, then $\varphi'(v) = 1$. Therefore Fitch's algorithm and Sankoff's algorithm will agreed on $\varphi'(v)$ if they agree on $\varphi'(\rho)$. □

Proof of Lemma 8.4. In each algorithm, once $\varphi(\rho)$ has been set, the algorithm deterministically outputs a most parsimonious labeling of all vertices. Therefore, it suffices to prove that both algorithms have the same choices for labeling the root and both algorithms output the same most parsimonious labeling for the same choice for $\varphi'(\rho)$.

If $B(\rho) = \{0\}$ or $B(\rho) = \{1\}$, then there is only one choice in Fitch's algorithm for $\varphi'(\rho)$. By Claim 8.5, Sankoff's algorithm has the same determined value for $\varphi'(\rho)$. Further, all vertices $v \in V(T)$ will have either $B(v) = \{0\}$ or $B(v) = \{1\}$ by condition (33) and Claim 8.7 completes the proof.

If $B(\rho) = \{0,1\}$, then in Fitch's algorithm, there are two choices for $\varphi'(\rho)$. By Claim 8.6, $s_0(\rho) = s_1(\rho)$ in Sankoff's algorithm, which means there are also two choices for $\varphi'(\rho)$. Claim 8.8 implies that if we make the same choice for the root, both algorithms give the same most parsimonious labeling φ'.

Sankoff's algorithm is guaranteed to find all most parsimonious labelings and the most parsimonious labelings from Fitch's algorithm coincide with those from Sankoff's algorithm, this implies that Fitch's algorithm finds all most parsimonious labelings. □

As mentioned earlier, these algorithms are designed for a tree T with leaf-labeling $\varphi : L(T) \to \{0,1\}$. However, given a tree T with leaf-labeling $\phi : L(T) \to \{0,1\}^n$, we can restrict all strings to a single coordinate and run one of the above algorithms to find a most parsimonious labelings for $V(T)$ in that coordinate. Repeat this for each coordinate. The most parsimonious labelings found for each coordinate can then be combined into a most parsimonious labeling of $V(T)$ that extends ϕ.

9. Complexity result for #Binary

Here is our main result on binary trees.

Theorem 9.1. *#Binary is #P-complete.*

Proof. A proof similar to that of Lemma 2.13 shows that #Binary is in #P. To prove #Binary is in #P-hard, we provide a polynomial reduction from #D3SAT.

Fix a D3CNF, $\Gamma = \bigwedge_{i\in[k]} c_i$ with k clauses and n variables. Let

$$\Psi(\Gamma) := \bigwedge_{i\in[k]} c_i \wedge \bigwedge_{i\in[n]} \Phi_i$$

with $2n$ variables, $\{v_1, v_2, \dots v_n, w_1, w_2, \dots, w_n\}$, where each clause c_i has three distinct literals from $\{v_i, \overline{v_i} : i \in [n]\}$, and Φ_i is the D3CNF in (24) which guarantees that, for each $i \in [n]$, v_i and w_i have different truth values in a satisfying assignment. By Lemma 8.3, there is a bijection between the satisfying truth assignments for Γ and the satisfying truth assignments for $\Psi(\Gamma)$. We will construct a binary tree $\mathcal{B}$ and define a labeling φ of its leaves with binary strings in B such that the number of satisfying truth assignment for $\Psi(\Gamma)$ is directly computable from $Z_{T,\varphi}(B, x!)$, the number of most parsimonious scenarios for binary tree T with leaf labeling φ.

Each Φ_i has 4 clauses, so $\Psi(\Gamma)$ has $k + 4n$ clauses. Assign the names $c_{k+1}, \dots, c_{k+4n}$ to the $4n$ clauses of $\bigwedge_{i\in[n]} \Phi_i$. Then

$$\Psi(\Gamma) = \bigwedge_{i\in[k+4n]} c_i.$$

For each $i \in [k + 4n]$, we define a binary tree $\mathcal{B}_i$ which encodes clause c_i. The final binary tree $\mathcal{B}$ will join $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_{k+4n}$ by a comb. For $t = 148(16n^2 + 8kn)(k + 4n)$, the leaf-labeling $\varphi : L(\mathcal{B}) \to \{0, 1\}^{2n+t}$, will assign a binary string with coordinates $(x_1, y_1, \dots, x_n, y_n, e_1, \dots, e_t)$ to each leaf. The x_i coordinates will correspond to the v_i variables and the y_i coordinates will correspond to the w_i variables of $\Psi(\Gamma)$. The e_i coordinates will be for additional ones, used in a manner similar to the additional ones in the previous sections for star trees.

In this section and the next, we denote the left child of a non-leaf vertex v by v_ℓ and the right child by v_r. The height of a vertex is its graph distance from the root. The construction of $\mathcal{B}_i$ with its leaf labeling φ will come in Definition 9.5, but first we need some preliminary definitions.

For any clause $c = v_\alpha \vee v_\beta \vee v_\gamma$ which is the disjunction of 3 distinct literals, Miklós, Kiss, and Tannier (2014) defined a *unit subtree*, $\mathcal{U}$, with 248 leaves. They also defined a leaf-labeling $\hat{\varphi} : L(\mathcal{U}) \to \{0, 1\}^{151}$ where the binary strings in the range have coordinates $(\hat{x}_\alpha, \hat{x}_\beta, \hat{x}_\gamma, \hat{e}_1, \hat{e}_2, \dots, \hat{e}_{148})$. The first three coordinates correspond to the variables in c_i and the remaining 148 coordinates are for additional ones. This unit subtree has some useful properties which will be discussed after Definition 9.5.

For each $i \in [k + 4n]$, let $\mathcal{U}_i$ be the unit subtree for clause c_i. If $i \le k$ where c_i relates $v_\alpha, v_\beta, v_\gamma$, then $\mathcal{U}_i$ will have leaf labels with coordinates $\{x_\alpha, x_\beta, x_\gamma\}$ and 148 coordinates for additional ones. If $i > k$ where c_i relates variables $v_\alpha, w_\alpha, v_{\alpha+1}$, then $\mathcal{U}_i$ will have leaf labels with coordinates $\{x_\alpha, y_\alpha, x_{\alpha+1}\}$ and 148 coordinates for additional ones.

Definition 9.2. *The tree $\mathcal{T}_i$ in Step 5 of Definition 9.4 is a comb joining $16n^2 + 8kn$ copies of $\mathcal{U}_i$, as in Figure 1.*

Definition 9.3. *For three literals a, b, c, we define $S(a, b, c)$ to be the complete binary tree of height 3 with root ρ with the vertices labeled with equations as follows:*

1. *Assign the label "$a = 0$" to vertex ρ_ℓ and "$a = 1$" to ρ_r.*
2. *For each vertex u of height 1, assign the label "$b = 0$" to u_ℓ and "$b = 1$" to u_r.*
3. *For each vertex v of height 2, assign the label "$c = 0$" to v_ℓ and "$c = 1$" to v_r.*

This tree is pictured on the right in Figure 2. We will use the representation on the left in place of $S(a, b, c)$ in future figures.

Next we construct $\hat{\mathcal{B}}_i$ which will have the same tree structure as the desired $\mathcal{B}_i$. However, $\hat{\mathcal{B}}_i$ will have all of its vertices labeled with equations while $\mathcal{B}_i$ will only have leaf labels which are binary strings. The leaf

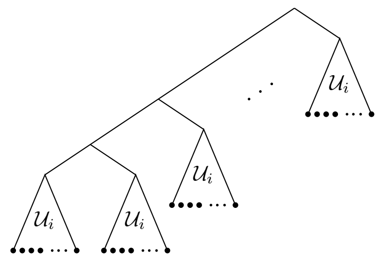

FIGURE 1. Comb connecting $16n^2 + 8kn$ copies of $\mathcal{U}_i$ to create $\mathcal{T}_i$

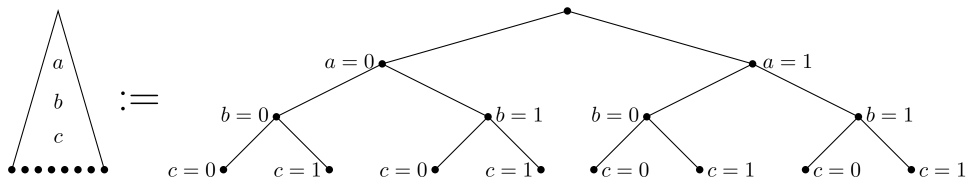

FIGURE 2. The labeled binary tree on the right is $S(a, b, c)$. The representation on the left will be used in place of $S(a, b, c)$ in future figures.

labeling of $\mathcal{B}_i$ will be induced by the vertex labels of $\hat{\mathcal{B}}_i$. Each leaf will essentially inherit the labels of its ancestors.

Definition 9.4. *Fix $i \in [k + 4n]$. Construct $\hat{\mathcal{B}}_i$, a binary tree with vertex labels, as follows.*

A. If $i \in [k]$, then say clause c_i has variables $v_\alpha, v_\beta, v_\gamma$. The construction of $\hat{\mathcal{B}}_i$ described below is drawn in Figure 3.
   - *(a) Draw a vertex ρ^i with two children, ρ^i_ℓ and ρ^i_r.*
   - *(b) Label vertex ρ^i_ℓ with the equations "$x_j = y_j = 0$" for each $j \in [n] \setminus \{\alpha, \beta, \gamma\}$. Label ρ^i_r with "$x_j = y_j = 1$" for all $j \in [n] \setminus \{\alpha, \beta, \gamma\}$.*
   - *(c) From each of ρ^i_ℓ and ρ^i_r, hang a copy of $S(y_\alpha, y_\beta, y_\gamma)$.*
   - *(d) From each leaf of each copy of $S(y_\alpha, y_\beta, y_\gamma)$, hang a copy of $S(x_\alpha, x_\beta, x_\gamma)$.*
   - *(e) Delete the left-most copy of $S(x_\alpha, x_\beta, x_\gamma)$, the one which hangs below the vertices with labels "$y_\alpha = 0$," "$y_\beta = 0$," "$y_\gamma = 0$," and with ancestor ρ^i_ℓ, and replace it with a copy of the comb $\mathcal{T}_i$ from Definition 9.2.*

B. If $i \in \{k + 1, \ldots, k + 4n\}$, then clause c_i relates variables $v_\alpha, w_\alpha, v_{\alpha+1}$. The construction of $\hat{\mathcal{B}}_i$ described below requires only a change of variables from the previous construction.
   - *(a) Draw a vertex ρ^i with two children, ρ^i_ℓ and ρ^i_r.*
   - *(b) Label ρ^i_ℓ with the system of equations "$x_j = y_j = 0$" for all $j \in [n] \setminus \{\alpha, \alpha + 1, \alpha + 2\}$. Label ρ^i_r with equations "$x_j = y_j = 1$" for each $j \in [n] \setminus \{\alpha, \alpha + 1, \alpha + 2\}$.*
   - *(c) From each of ρ^i_ℓ and ρ^i_r, hang a copy of $S(y_{\alpha+1}, x_{\alpha+2}, y_{\alpha+2})$.*
   - *(d) Hang a copy of $S(x_\alpha, y_\alpha, x_{\alpha+1})$ from each leaf of each and every copy of $S(y_{\alpha+1}, x_{\alpha+2}, y_{\alpha+2})$.*
   - *(e) Delete the left-most copy of $S(x_\alpha, y_\alpha, x_{\alpha+1})$ and replace it with a copy of comb $\mathcal{T}_i$ from Definition 9.2.*

Recall $\mathcal{B}$ is a comb connecting binary trees $\mathcal{B}_i$. The binary tree $\mathcal{B}$ has a leaf labeling $\varphi : L(\mathcal{B}) \to \{0, 1\}^{2n+t}$ where

$$t = 148(16n^2 + 8kn)(k + 4n).$$

Each leaf label will have coordinates $(x_1, y_1, \ldots, x_n, y_n, e_1, \ldots, e_t)$. In the next definition, we define $\mathcal{B}_i$ and values of φ on the leaves of $\mathcal{B}_i$.

Definition 9.5. *For each $i \in [k + 4n]$, the binary tree $\mathcal{B}_i$ will have the same tree structure as $\hat{\mathcal{B}}_i$. We only need to explain the labeling $\varphi : L(\mathcal{B}_i) \to \{0, 1\}^{2n+t}$.*

Partition $[t]$ into classes E_{ij} with $|E_{ij}| = 148$ for each $i \in [k + 4n]$ and each $j \in [16n^2 + 8kn]$. Identify the set E_{ij} with the j^{th} copy of $\mathcal{U}_i$ in $\mathcal{T}_i$. Here we define $\varphi(\ell)$ for each leaf $\ell \in L(\mathcal{B}_i)$.

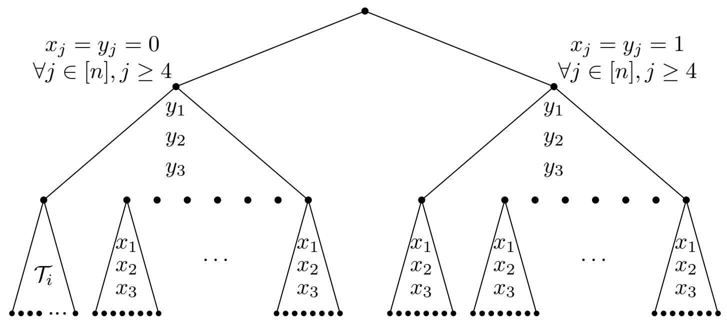

FIGURE 3. The binary tree $\hat{\mathcal{B}}_i$, for $i \in [k]$ created for clause $c_i = x_1 \vee x_2 \vee x_3$.

There are two cases:

- *If leaf ℓ is not in subtree $\mathcal{T}_i$, then $\varphi(\ell)[e_s] = 0$ for all $s \in [t]$. The value of each $\varphi(\ell)[x_w]$ and $\varphi(\ell)[y_w]$ for $w \in [n]$ is inherited from the labels of the ancestors of ℓ as they appeared in $\hat{\mathcal{B}}_i$.*
- *If leaf ℓ is in the subtree $\mathcal{T}_i$ within $\hat{\mathcal{B}}_i$, then it is a leaf within the j^{th} copy of unit subtree $\mathcal{U}_i$ for some $j \in [16n^2 + 8kn]$. Recall $\hat{\varphi}(\ell) \in \{0,1\}^{151}$. Identify the coordinates $\{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_{148}\}$ with the indices in the 148 coordinates of E_{ij} in any order. If $\hat{e}_s$ corresponds to coordinate e_r for $r \in E_{ij}$, then we require $\varphi(\ell)[e_r] = \hat{\varphi}(\ell)[\hat{e}_s]$. If z is one of the three coordinates which correspond to a variable in c_i, we also require that $\varphi(\ell)[z] = \hat{\varphi}(\ell)[\hat{z}]$. Set $\varphi(\ell)[e_s] = 0$ for $s \notin E_{ij}$. All other coordinates of $\varphi(\ell)$ will take the value 0 (the value inherited from the labeling of their ancestors in $\hat{\mathcal{B}}_i$).*

Define $\varphi_{x_j} : L(\mathcal{B}) \to \{0,1\}$ so that $\varphi_{x_j}(\ell) = \varphi(\ell)[x_j]$. Define φ_{y_j} and φ_{e_j} similarly. We want to examine the Fitch solutions on $\mathcal{B}$ for each φ_{x_j}, φ_{y_i}, and φ_{e_j}. We will first prove that the conditions of Lemma 8.4 hold and thus Fitch's algorithm find all most parsimonious labelings for φ on $\mathcal{B}$.

We first explore the Fitch solutions for φ_{e_j}, $j \in [t]$.

Fact 9.6. *Fix $j \in [t]$. There is only one $\ell \in L(\mathcal{B})$ with $\varphi_{e_j}(\ell) = 1$. After running Part 1 of Fitch's algorithm, $B(\ell) = \{1\}$, the parent v of ℓ has $B(v) = \{0,1\}$, and $B(u) = \{0\}$ for all other vertices. Consequently, Part 2 of Fitch's algorithm will output a most parsimonious labeling φ'_{e_j} such that $\varphi'_{e_j}(\ell) = 1$ and for all other vertices $u \in V(\mathcal{B})$, $\varphi'_{e_j}(u) = 0$.*

Proof. These values of B follow directly from the description of $\varphi(\ell)[e_j]$, for leaf ℓ, which was given in Definition 9.5. The conclusion follows from the definition of φ' (27). □

Fact 9.7. *For $j \in [t]$, there is only one most parsimonious labeling φ'_{e_j} which extends leaf labeling φ_{e_j} of $\mathcal{B}$.*

Proof. Recall that most parsimonious labelings minimize the sum of Hamming distances between adjacent vertices in the tree. The most parsimonious labeling obtained from Fitch's algorithm has

$$\sum_{uv \in E(\mathcal{B})} H(\varphi'_{e_j}(u), \varphi'_{e_j}(v)) = 1.$$

Because there is only one leaf ℓ with $\varphi_{e_j}(\ell) = 1$, the φ'_{e_j} obtained from Fitch's algorithm is the only extension of φ_{e_j} with the sum of Hamming distances equal 1. □

Fix $j \in [n]$. Next we consider the most parsimonious labelings for φ_{x_j} on $\mathcal{B}$. The same arguments will hold for each φ_{y_j}.

Run Part 1 of Fitch's algorithm on $\mathcal{B}$ with leaf labeling φ_{x_j}. For those clauses c_i which contain variable v_j, we have the following result.

Proposition 9.8 (Miklós, Kiss, and Tannier 2014)**.** *Fix a clause c_i. Suppose variable v_j is in c_i with coordinate x_j corresponding to variable v_j. Let r^i be the root of unit subtree $\mathcal{U}_i$ for c_i. Run Fitch's algorithm on $\mathcal{U}_i$ with leaf labeling φ_{x_j}. The following hold:*

(1) $B(r^i) = \{0,1\}$.
(2) For $u, v \in V(U_i)$, if v is a child of u, then $B(v) = \{0,1\} \Rightarrow B(u) = \{0,1\}$.

In a single copy of $S(a,b,c)$, all vertices of the same distance from the root either have $B(v) \in \{\{0\},\{1\}\}$ or all of them have $B(v) = \{0,1\}$. This fact together with Proposition 9.8 implies that, when Fitch's algorithm is run on $\mathcal{B}$ with leaf-labeling $\varphi(x_i)$, for any $u, v \in V(T)$ with v a child of u,

$$B(v) = \{0,1\} \Rightarrow B(u) = \{0,1\}.$$

With this result and the structure of each $S(a,b,c)$, by Lemma 8.4, we can conclude that Fitch's algorithm finds all most parsimonious labelings of $\mathcal{B}$ that extend φ_{x_j}. Further, $B(\rho) = \{0,1\}$ implies there are exactly two such most parsimonious labelings.

As mentioned earlier, these results also hold for coordinate y_i. Fitch's algorithm finds the only two most parsimonious labelings that extend φ_{y_j} on $\mathcal{B}$.

For most parsimonious labeling φ' that extends φ, on each $v \in V(T)$, notate $\varphi'(v)[x_j]$ by φ'_{x_j}. Likewise, define the notations φ'_{y_j} and φ'_{e_s}.

Lemma 9.9. *For leaf labeling φ of $\mathcal{B}$, Fitch's algorithm finds all most parsimonious labelings. Each is characterized by the string it assigns to the root ρ of $\mathcal{B}$ and there are precisely 2^{2n} most parsimonious labelings, one for each root label in $\{0,1\}^{2n} \times \{0\}^t$.*

Proof. Given a most parsimonious labeling φ' that extends φ, each φ'_{x_j}, φ'_{y_j}, and φ'_{e_s} is a most parsimonious labeling for that coordinate. So it suffices to first find all most parsimonious scenarios for the leaf labelings $\varphi_{x_j}, \varphi_{y_j}, \varphi_{e_s}$ for all $j \in [n]$ and $s \in [t]$ and take combinations of these labelings.

We have already seen that Fitch's algorithm will find all most parsimonious labelings for φ_{x_j} and φ_{y_j}, and there are exactly 2 of each. Fitch's algorithm will also find the one and only most parsimonious labeling for φ_{e_s}. Therefore, there are 2^{2n} most parsimonious labelings of $\mathcal{B}$ that extend φ. Part 2 of Fitch's algorithm shows that each most parsimonious labeling is characterized by the string it assigns to ρ. Since $B(\rho) = \{0,1\}$ for each φ_{x_j} and φ_{y_j} and $B(\rho) = \{0\}$ for each φ_{e_s}, the possible strings for $\varphi'(\rho)$ are $\{0,1\}^{2n} \times \{0\}^t$. $\square$

Set $\mathcal{M} := \{0,1\}^{2n} \times \{0\}^t$.

Definition 9.10. *There is a bijection between $\mathcal{M}$ and the possible truth assignments for $\Psi(\Gamma)$. In particular, given any $\mu \in \mathcal{M}$, define a truth assignment for variables $\{v_i\}_{i=1}^n \cup \{w_i\}_{i=1}^n$ as follows:*

- *For each $i \in [n]$, let v_i be assigned the value true if $\mu[x_i] = 1$ and false otherwise.*
- *For each $i \in [n]$, let w_i be assigned the value true if $\mu[y_i] = 1$ and false otherwise.*

Define $\mathcal{M}_{\Psi(\Gamma)}$ to be the set of $\mu \in \mathcal{M}$ which correspond to satisfying truth assignments for $\Psi(\Gamma)$. Likewise, for any Θ, a clause or conjunction of clauses from $\Psi(\Gamma)$, define $\mathcal{M}_\Theta$ to be the set of $\mu \in \mathcal{M}$ which correspond to satisfying truth assignments for Θ.

Now we know that each most parsimonious labeling of $\mathcal{B}$ extending φ is found using Fitch's algorithm and is characterized by the binary string it assigns to the root. From here, we are interested in the number of scenarios admitted by each of these most parsimonious labelings. Ultimately, we wish to make a distinction between the binary strings in $\mathcal{M}_{\Psi(\Gamma)}$ and those in $\mathcal{M} \setminus \mathcal{M}_{\Psi(\Gamma)}$ by examining the number of scenarios admitted by the corresponding most parsimonious labeling.

Let φ' be a most parsimonious labeling for $\mathcal{B}$. The number of scenarios which are admitted by φ' is precisely

$$\mathcal{H}(\varphi'(\rho)) := \prod_{uv \in E(\mathcal{B})} H(\varphi'(u), \varphi'(v))!.$$

To calculate this, we partition the edges of $\mathcal{B}$ into 4 sets.

First, consider the edges of the comb which connects $\{\mathcal{B}_i\}_{i=1}^{k+4n}$ to form $\mathcal{B}$. Part 2 of Fitch's algorithm will set $\varphi'(\rho) = \varphi'(\rho^i)$ where ρ^i is the root of $\mathcal{B}_i$. So the Hamming distance along each of these edges is 0.

Next we look within each $\mathcal{B}_i$.

Claim 9.11. *Set $\Phi := \bigwedge \Phi_\iota$ as defined in* (25). *For $i \in [k+4n]$, let ρ^i be the root of $\mathcal{B}_i$ with children ρ^i_ℓ and ρ^i_r. Set $\eta := \varphi'(\rho^i)$. If $\eta \in \mathcal{M}'_\Phi$, then*

$$H(\eta, \varphi'(\rho^i_\ell)) = H(\eta, \varphi'(\rho^i_r)) = n - 3.$$

Otherwise

$$(n-3)!^2 \le H(\eta, \varphi'(\rho^i_\ell))! \cdot H(\eta, \varphi'(\rho^i_r))! \le (2n-6)!0!.$$

Proof. Suppose $\eta \in \mathcal{M}'_\Phi$. Then for each $j \in [n]$ considered in Step 2 of Definition 9.4 (there are $n-3$ such j) , if $\eta[x_j] = 0$ then we have the following properties:

- $\eta[y_j] = 1$ because η corresponds to a satisfying assignment for Φ,
- $0 = \eta[x_j] \neq \varphi'(\rho_r^i)[x_j] = 1$,
- $1 = \eta[y_j] \neq \varphi'(\rho_\ell^i)[y_j] = 0$.

On the other hand, if $\eta[x_j] = 1$ then we have the following properties:

- $\eta[y_j] = 0$ because η corresponds to a satisfying assignment for Φ,
- $1 = \eta[x_j] \neq \varphi'(\rho_\ell^i)[x_j] = 0$,
- $0 = \eta[y_j] \neq \varphi'(\rho_r^i)[y_j] = 1$.

For each $s \in [t]$, $\eta[e_s] = \varphi'(\rho_\ell^i)[e_s] = \varphi'(\rho_r^i)[e_s] = 0$. For each $j \in [n]$ which was not considered in Step 2 of Definition 9.4, $\eta[x_j] = \varphi'(\rho_\ell^i)[x_j] = \varphi'(\rho_r^i)[x_j]$ and $\eta[y_j] = \varphi'(\rho_\ell^i)[y_j] = \varphi'(\rho_r^i)[y_j]$ because the B values (from Fitch's algorithm) for these coordinates at these vertices will be $\{0,1\}$. Thus

$$H(\eta, \varphi'(\rho_\ell^i)) = H(\eta, \varphi'(\rho_r^i)) = n-3.$$

Alternatively, if $\eta \notin \mathcal{M}_\Phi$, then $H(\eta, \varphi'(\rho_\ell^i)) + H(\eta, \varphi'(\rho_r^i)) = 2n-6$ because each x_i and each y_i will contribute 1 to one of the Hamming distances. Using the convexity of the factorial, this establishes the last line of the claim. ☐

Based on the construction of $S(a,b,c)$, the Hamming distance $H(\varphi'(u), \varphi'(v))$ for each edge uv in each copy of $S(a,b,c)$ is exactly 1.

The only piece remaining is $\mathcal{T}_i$. We make the following remarks for the clause $c_i = v_1 \vee v_2 \vee v_3$ to make the explanation easier. However, the arguments can be extended for any clause c_i in $\Psi(\Gamma)$.

Fact 9.12. *If t^i is the root of $\mathcal{T}_i$ and r^i is the root of one of the copies of $\mathcal{U}_i$ below $\mathcal{T}_i$, then running Fitch's algorithm for each coordinate, we find*

- $B(t^i) = B(r^i) = \{0,1\}$ *for each x_i, $i \in [3]$, by Proposition 9.8.*
- $B(t^i) = B(r^i) = \{0\}$ *for each x_i, $i \geq 4$, by the construction of $\mathcal{B}_i$.*
- $B(t^i) = B(r^i) = \{0\}$ *for each y_i, $i \in [n]$, by the construction of $\mathcal{B}_i$.*
- $B(t^i) = B(r^i) = \{0\}$ *for each e_s, $s \in [t]$, because there is only one leaf $\ell \in L(\mathcal{B})$ with $\varphi(\ell)[e_s] = 1$.*

Therefore, it is easy to see that, along the edges of the comb which connect the copies of $\mathcal{U}_i$, the Hamming distances will be 0.

Next we turn our attention to a single copy of $\mathcal{U}_i$, say the j^{th} copy.

Fact 9.13. *Fix Γ and build binary tree $\mathcal{B}$. Fix a most parsimonious labeling φ' which extends leaf labeling φ. For clause $c_i = v_1 \vee v_2 \vee v_3$, we have the following characteristics for each $v \in \mathcal{U}_i$,*

- *for $s \geq 4$, $\varphi'(v)[x_s] = 0$,*
- *for $s \in [n]$, $\varphi'(v)[y_s] = 0$,*
- *for $s \notin E_{ij}$, $\varphi'(v)[e_s] = 0$.*

Therefore, only the values of $\varphi'(v)$ on the coordinates x_1, x_2, x_3 and e_s for $s \in E_{ij}$ will affect the Hamming distances along the edges in $\mathcal{U}_i$. These are precisely the 151 coordinates that appeared in the original labeling $\hat{\varphi}$ of the leaves of $\mathcal{U}_i$ given by Miklós, Kiss, and Tannier (2014). For each $v \in V(\mathcal{U}_i)$, define $\hat{\varphi}'(v) : V(\mathcal{U}_i) \to \{0,1\}^{151}$ to be the restriction of $\varphi'(v)$ to these 151 coordinates. In particular, $\hat{\varphi}'$ is a most parsimonious labeling on $\mathcal{U}_i$ which extends leaf labeling $\hat{\varphi}$.

The following fact is a consequence of Fact 9.13.

Fact 9.14. *Let r^i be the root of $\mathcal{U}_i$. If $\varphi'(r^i) = \hat{\varphi}'(r^i)$, then for each $uv \in E(\mathcal{U}_i)$,*

$$H(\varphi'(u), \varphi'(v)) = H(\hat{\varphi}'(u), \hat{\varphi}'(v)).$$

As a result

$$\prod_{uv \in \mathcal{U}_i} H(\varphi'(u), \varphi'(v))! = \prod_{uv \in \mathcal{U}_i} H(\hat{\varphi}'(u), \hat{\varphi}'(v))!.$$

This is calculated as follows:

Fact 9.15 (Miklós, Kiss, and Tannier 2014)**.** *Fix $i \in [k+4n]$, the binary tree $\mathcal{U}_i$ with root r^i, and leaf-labeling $\hat{\varphi}$. Then for any most parsimonious labeling $\hat{\varphi}'$ which extends $\hat{\varphi}$:*

(1) If $\hat{\varphi}'(r^i)$ corresponds to a satisfying truth assignment for c_i, then

$$\prod_{uv\in\mathcal{U}_i} H(\hat{\varphi}'(u),\hat{\varphi}'(v))! = 2^{156}\times 3^{64}.$$

(2) If $\hat{\varphi}'(r^i)$ corresponds to a truth assignment which does not satisfy c_i, then

$$\prod_{uv\in\mathcal{U}_i} H(\hat{\varphi}'(u),\hat{\varphi}'(v))! = 2^{136}\times 3^{76}.$$

Since $\varphi'(\rho)=\varphi'(r^i)=\hat{\varphi}'(r^i)$, $\varphi'(\rho)$ corresponds to a satisfying truth assignment for c_i if and only if $\hat{\varphi}'(r^i)$ also corresponds to a satisfying truth assignment for c_i.

As a result of the above discussion, we have proven the following claim.

Claim 9.16. *Fix $i\in[k+4n]$. If $\varphi'(\rho)$ corresponds to a satisfying truth assignment for clause c_i and $\bigwedge_{\iota\in[n]}\Phi_\iota$, then*

$$\prod_{uv\in E(\mathcal{B}_i)} H(\varphi'(u),\varphi'(v))! = (n-3)!^2\left(2^{156}\times 3^{64}\right)^{16n^2+8kn}.$$

If $\varphi'(\rho)$ corresponds to a truth assignment which does not satisfy c_i, then

$$\begin{aligned}(n-3)!^2\left(2^{136}\times 3^{76}\right)^{16n^2+8kn} &\le \prod_{uv\in E(\mathcal{B}_i)} H(\varphi'(u),\varphi'(v))!\\ &\le (2n-6)!\left(2^{136}\times 3^{76}\right)^{16n^2+8kn}.\end{aligned}$$

If $\varphi'(\rho)$ corresponds to a truth assignment which satisfies c_i but does not satisfy $\bigwedge_{i\in[n]}\Phi_i$, then

$$\begin{aligned}(n-3)!^2\left(2^{156}\times 3^{64}\right)^{16n^2+8kn} &< \prod_{uv\in E(\mathcal{B}_i)} H(\varphi'(u),\varphi'(v))!\\ &\le (2n-6)!\left(2^{156}\times 3^{64}\right)^{16n^2+8kn}.\end{aligned}$$

Observe,

$$\begin{aligned}\frac{(2n-6)!\left[2^{136}\times 3^{76}\right]^{16n^2+8kn}}{(n-3)!^2\left[2^{156}\times 3^{64}\right]^{16n^2+8kn}} &= \left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn}\binom{2n-6}{n-3}\\ &< \left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn} 2^{2n}\\ &< \left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn} 2^{2n+k}\\ &= \left[\frac{3^{12}}{2^{20-1/(8n)}}\right]^{16n^2+8kn}\\ &< \left[\frac{3^{12}}{2^{19.5}}\right]^{16n^2+8kn}\\ &< 1.\end{aligned}$$

Consequently,

$$\begin{aligned}(2n-6)!\left(2^{136}\times 3^{76}\right)^{16n^2+8kn} &< (n-3)!^2\left(2^{156}\times 3^{64}\right)^{16n^2+8kn}\\ &< (2n-6)!\left(2^{156}\times 3^{64}\right)^{16n^2+8kn}.\end{aligned}$$

Claim 9.17. *If $\varphi'(\rho)$ corresponds to a satisfying truth assignment for $\Psi(\Gamma)$, then*

$$\mathcal{H}(\varphi'(\rho)) = \left[(n-3)!^2\left(2^{136}\times 3^{76}\right)^{16n^2+8kn}\right]^{k+4n} =: B_{good}.$$

If $\varphi'(\rho)$ corresponds to a truth assignment which does not satisfy $\Psi(\Gamma)$, then there must be a clause c_i for some $i \in [k+4n]$ which is not satisfied. Therefore,

$$\begin{aligned}\mathcal{H}(\varphi'(\rho)) \le& (2n-6)!\left(2^{136}\times 3^{76}\right)^{16n^2+8kn} \\ &\cdot\left[(2n-6)!\left(2^{156}\times 3^{64}\right)^{16n^2+8kn}\right]^{k+4n-1} \\ =:& B_{bad}.\end{aligned}$$

Define

$$B_{total} := \sum_{\varphi'} \mathcal{H}(\varphi'(\rho))$$

which is the total number of most parsimonious scenarios for $\mathcal{B}$ which extend leaf labeling φ, as in Definition 2.12.

Given only B_{total}, we would like to determine the number of satisfying truth assignments, $|S|$, for $\Psi(\Gamma)$.

$$\begin{aligned}B_{total} &= \sum_{\eta\in\mathcal{M}'_{\Psi(\Gamma)}} \mathcal{H}(\eta) + \sum_{\eta'\in\mathcal{M}\setminus\mathcal{M}'_{\Psi(\Gamma)}} \mathcal{H}(\eta') \\ &= |S|B_{good} + \sum_{\eta'\in\mathcal{M}\setminus\mathcal{M}'_{\Psi(\Gamma)}} \mathcal{H}(\eta').\end{aligned}$$

As long as $\sum_{\eta'\in\mathcal{M}\setminus\mathcal{M}'_{\Psi(\Gamma)}} \mathcal{H}(\eta') < B_{good}$, we can conclude that the number of satisfying truth assignments for $\Psi(\Gamma)$ (and for Γ) is precisely

$$\left\lfloor \frac{B_{total}}{B_{good}} \right\rfloor.$$

Observe, for $n \ge 2$,

$$\begin{aligned}\frac{2^{2n}B_{bad}}{B_{good}} &= 2^{2n}\left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn}\binom{2n-6}{n-3}^{k+4n} \\ &< 2^{2n}\left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn} 2^{2n(k+4n)} \\ &< 2^{8n^2+2kn+2n}\left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn} \\ &< 2^{8n^2+4kn}\left[\frac{3^{12}}{2^{20}}\right]^{16n^2+8kn} \\ &= \left[\frac{3^{12}}{2^{20-1/2}}\right]^{16n^2+8kn} \\ &< 1.\end{aligned}$$

Because there are only 2^{2n} truth assignments and 2^{2n} most parsimonious labelings, we obtain our desired result:

$$\sum_{\eta'\in U}\mathcal{H}(\eta') \le \sum_{\eta'\in U} B_{bad} \le 2^{2n}B_{bad} < B_{good}.$$

Therefore, if we could determine the total number of most parsimonious scenarios for this binary tree in polynomial time, then we could obtain the total number of satisfying assignments for $\Psi(\Gamma)$ and for Γ in polynomial time. This completes the proof. □

10. Conclusion

We proved that it is #P-complete to calculate the partition function $Z(B, x!)$. However, the existence of an FPAUS for this quantity has not yet been established. Following a number of results relating to calculating $Z(B, f(x))$ exactly for various functions $f(x)$, we where able to prove that, when $\log f(x)$ is strictly decreasing, under mild conditions an FPAUS exists for $Z(B, f(x))$ only if RP=NP. The question of approximating $Z(B, f(x))$ when $\log f(x)$ is strictly increasing remains unsettled. We concluded with a #P-complete result for the extension of the partition function to binary trees, a natural extension to the bioinformatics interpretation of $Z(B, x!)$.

11. Acknowledgements

Heather Smith acknowledges support from DARPA and AFOSR under contract #FA9550-12-1-0405, NSF DMS contracts 1300547 and 1344199, and a SPARC Graduate Research Grant from the Office of the Vice President for Research at the University of South Carolina.

References

Bach, E. and J. Shallit (1996). *Algorithmic Number Theory, Volume I: Efficient Algorithms.* MIT Press.

Bremaud, P. (2008). *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues (Texts in Applied Mathematics).* Springer.

Brightwell, G. and P. Winkler (1991). "Counting linear extensions." In: *Order* 8, pp. 225–242.

Cook, S. A. (1971). "The Complexity of Theorem-proving Procedures." In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing.* STOC '71. New York, NY, USA: ACM, pp. 151–158.

Erdős, P. L. and L. A. Székely (1994). "On weighted multiway cuts in trees." In: *Mathematical Programming* 65 (1-3), pp. 93 –105.

Feijão, P. and J. Meidanis (2011). "SCJ: A Breakpoint-Like Distance that Simplifies Several Rearrangement Problems." In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8.5, pp. 1318–1329.

Felsenstein, J. (2003). *Inferring Phylogenies.* Sinauer Associates.

Fitch, W. M. (1971). "Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology." In: *Systematic Zoology* 20.4, pp. 406–416. ISSN: 00397989.

Gill, J. (1977). "Computational complexity of probabilistic Turing machines." In: *SIAM Journal of Computing* 6.4, pp. 675–695.

Jerrum, M., L.G. Valiant, and V. V. Vazirani (1986). "Random generation of combinatorial structures from a uniform distribution". In: *Theoretical Computer Science* 43, pp. 169 –188.

Karzanov, A. and L. Khachiyan (1991). "On the conductance of order Markov chains." In: *Order* 8.1, pp. 7–15.

Metropolis, N. et al. (1953). "Equations of state calculations by fast computing machines." In: *Journal of Chemical Physics* 21.6, pp. 1087–1092.

Miklós, I., Sándor Z. Kiss, and E. Tannier (2014). "Counting and sampling SCJ small parsimony solutions." In: *Theoretical Computer Science* 552, pp. 83 –98.

Papadimitriou, C. (1994). *Computational Complexity.* Addison-Wesley.

Rosser, B. (1941). "Explicit bounds for some functions of prime numbers." In: *American Journal of Mathematics* 63.1, pp. 211–232.

Sankoff, D. and P. Rousseau (1975). "Locating the vertices of a Steiner tree in an arbitrary metric space." In: *Mathematical Programming* 9.1, pp. 240–246. ISSN: 0025-5610. DOI: 10.1007/BF01681346.

Valiant, L.G. (1979). "The complexity of computing the permanent." In: *Theoretical Computer Science* 8.2, pp. 189 –201.

Welsh, D. (1993). *Complexity: Knots, Colourings and Countings.* London Mathematical Society Lecture Note Series 186. Cambridge University Press.